

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

PC

特集 マシン語への第一歩

S-OSリロケータブルアセンブラWZD
ハードウェア工作入門/PurePASCAL
ノーマルX1対応コマンドシェルINTEGRAL X1

7
1990

**SOFT
BANK**

オーノエックス
定価560円



SHARP



Δ68000 SX-WINDOW

ひらかれた知性。

見はてぬ夢の象徴。

次代のインテリジェンス、^{*}“SX-WINDOW”搭載。

*SX-WINDOWの起動には、メインメモリ2MBが必要です。PRO IIシリーズ (CZ-6530/6630)でSX-WINDOWをご使用の際は、あらかじめ別売の1MB増設RAMボードCZ-6BE1Bを増設してください。



●いま、こだわり続けてきたある執着がまさに帰結しようとしています。グラフィカルユーザーインターフェイス“SX-WINDOW ver.1.0”。もちろん、X68000には発売当初よりビジュアルシェルスが搭載されていたことはご存じのとおりですが、クオリティグラフィックやマルチメディア、マルチタスク対応など真の意味での汎用性を志向した開発コンセプトからは、私たち自身ものたりなさを禁じ得なかったことも事実です。しかし、キャラクタユーザーインターフェイス全盛のその時代に、デスクトップの概念をいち早く採り入れた先見性は、現在のインターフェイスの在り方に対する的確な予測に基づくもので、何よりも、トレンドなユーザーの圧倒的な支持によって証明されています。パーソナルコンピュータがその意味どおり、個人のためのツールなら、インターフェイスの発展は必然です。このウィンドウシステムは、私たちX68000開発プロジェクトに携わったすべてのスタッフの指標であり、義務でもあったのです——。

●ユーザー本位の操作環境を提供するフル画面マルチウィンドウタイプのデスクトップ(テキスト面/単色4階調+カラー4色。グラフィック面/カラー65,536色中16色)。新感覚スクロールバー……こだわりの美学で高められたユーザーインターフェイス。イベントドリブン型マルチタスク処理により複数の作業を同時に処理できる疑似マルチタスクや入出力装置の設定が簡単におこなえる多機能コントロールパネルを搭載した本格ウィンドウシステムです。

●“SX-WINDOW”。このひらかれた知性は、今もそしてすぐ後に続く時代をも含めた質の高い「愉しみ」を提供するインターフェイスです。フレンドリーOS Human68kはここに、当初の目的の成就と共に、将来へ確かな展望を明示したといえるでしょう。さまざまなジャンルへ、拡がりど密度を高めるアプリケーション環境、インテリジェントなペリフェラル環境。こうしたトレンドを背景に、いま第4世代のX68000がデビューします。



NEW

X68000

PERSONAL WORKSTATION

SUPER・EXPERT・PRO

ザ・ワークステーション。80Mバイトハードディスク、SCSI インターフェイスを標準装備。

SUPER HD 本体+キーボード+マウス+トラックボール

HDタイプ CZ-623C-TN(チタンブラック) 標準価格498,000円(税別) <6月発売予定>

アートの系譜。 **EXPERT II** 本体+キーボード+マウス+トラックボール

CZ-603C-BK(ブラック)・GY(グレー) 標準価格338,000円(税別)/HDタイプ CZ-613C-BK(ブラック) 標準価格448,000円(税別)

ニュースタンド。 **PRO II** 本体+キーボード+マウス

CZ-653C-BK(ブラック)・GY(グレー) 標準価格285,000円(税別)

HDタイプ CZ-663C-BK(ブラック)・GY(グレー) 標準価格395,000円(税別)

15型カラーディスプレイテレビ(ドットピッチ0.39mm)	OZ-602D-BK(ブラック)・GY(グレー).....	標準価格 99,800円(チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.39mm)	CZ-605D-BK(ブラック)・GY(グレー).....	標準価格115,000円(スピーカー2個/チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.31mm)	CZ-613D-TN(チタンブラック)・BK(ブラック)・GY(グレー).....	標準価格135,000円(スピーカー2個/チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31mm)	CZ-603D-BK(ブラック)・GY(グレー).....	標準価格 84,800円(チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31mm)	OZ-604D-BK(ブラック)・GY(グレー).....	標準価格 94,800円(スピーカー2個/チルトスタンド同梱・税別)

今回のゲームソフトウィルスの件では、ユーザー各位に心配をおかけ致しました。ワクテンソフトに関しては、お近くの弊社サービスセンターにお尋ね下さい。X68000及び関係商品は、これまで通り安心してご愛用頂けます。

EXEリターンズグッズ
プレゼント実施中

- いま、EXE会員よりご紹介のお客様がEXEショップでX68000シリーズを購入されたと、EXE会員にEXEリターンズグッズをプレゼントします。詳しくはEXEショップにお問い合わせください。
- また、X68000シリーズをご購入のお客様は、ぜひEXEクラブにご入会ください。

本広告に掲載しております商品および役務の価格には消費税は含まれておりませんので、ご購入の際、消費税額をお支払い下さい。

●お問い合わせは…シャープ ㈱ 電子機器事業本部システム機器営業部 〒545大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

電子機器事業本部液晶映像システム事業部第2商品企画部 〒162東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)

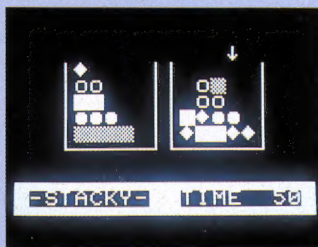
シャープ株式会社



マイクロコンピュータショー '90



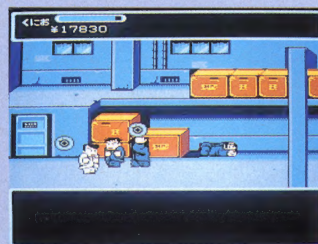
DōGA・CGアニメーション講座



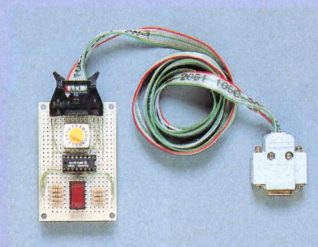
スタッキー



サーク



ダウントウン熱血物語



ハードウェア工作入門

Oh!X

C O N T

●特集

41 マシン語への第一歩

- 42 ぜんまいちゃん再び 吉田幸一
- 47 マシン語ってなあに? 毛内俊行
- 52 MC68000の動作を探る 宮島 靖
- 58 基本装備とおまじない 古村 聡
- 63 DOSコール&IOCSコールを使う 影山裕昭
- 67 マルチタスクへの挑戦 桑野雅彦
- 74 S-OS&REDAを忘れていませんか 山田純二

●カラー紹介

27 マイコンショウ'90&第70回ビジネスショウ

40 Oh!X Graphic Gallery DōGA・CGアニメーション

●THE SOFTOUCH

30 SOFTWARE INFORMATION 話題のソフトウェア

- 32 GAME REVIEW 西川善司
- 34 サーク 国津良男
- 36 あーくしゅ 山田純二
- 36 ダウントウン熱血物語

AFTER REVIEW 38 グラナダ/FAR SIDE MOON サーク/三国志 II

<スタッフ>

●編集長/前田 徹 ●編集/植木章夫 岡崎栄子 浅井研二 ●協力/有田隆也 中森 章 後藤貴行
林 一樹 荻窪 圭 岡本浩一郎 毛内俊行 吉田賢司 影山裕昭 相馬英智 古村 聡 村田敏幸 丹
明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 浦川博之 山田純二 ●カメラ/杉山和美 ●イラ
スト/永沢しげる 山田晴久 小栗由香 ●アートディレクター/島村勝頼 ●レイアウト/元木昌子
AD GREEN ●校正/千野延明 織田洋子



表紙絵：塚田 哲也

E N T S

●シリーズ全機種共通システム

77 THE SENTINEL

78 リロケータブルアセンブラWZD

石上達也

●読みのもの

134 猫とコンピュータ 第49回
ホットラインで

高沢恭子

153 X-OVER NIGHT 第2話
本命は恐ろしい

高原秀己

●連載/紹介/講座/プログラム

96 DōGA・CGアニメーション講座(11)
復活のCGアニメーション

かまたゆたか

105 ハードウェア工作入門(1)
基本インタフェイス回路 その1

三沢和彦

108 X-BASICプログラミング調理実習(12)
パズルゲームを作る(後)

泉 大介

114 PASCALプログラミングへの招待(2)
PASCALの特徴的な性格について

藤井義巳・藤木健士

117 マシン語カクテル in Z80's Bar 第13回
電卓はビットシフトで

山田純二

121 Oh!X LIVE in '90
夢幻戦士ヴァリスⅡよりSCARED SACREIFICE(X68000)
トッカータとフーガニ短調BWV565(X1/turbo)

渡辺一彦

花井章能

126 (で)のショートプロばーてい その11
前夜祭はすたっきいだ!

古村 聡

130 アフターケアのページ
創刊8周年記念PRO-68Kのその後

編集部

137 X1 turbo用ディスク管理プログラムINTEGRAL X1
ノーマルX1への対応

亀田雅彦

154 新製品速報
いよいよ発売開始! X68000SUPER-HD

丹 明彦

愛読者プレゼント.....152

ペンギン情報コーナー.....156

FILES Oh!X.....158

Oh!X質問箱.....160

STUDIO X.....162

編集部から/DRIVE ON/ごめんなさいのコーナー/SHIFT BREAK/microOdyssey.....166

1990 JUL. 7

UNIXはAT&T BELL LABORATORIESのOS名です。

Machはカーネギーメロン大学のOS名です。

CP/M, P-CPM, CP/M plus, CP/M-86, CP/M-68K, CP/M-80

00, DR-DOSはDIGITAL RESEARCH

OS/2はIBM

MS-DOS, MS-OS/2, XENIX, MACRO 80, MS CはMICROSOFT

MSX-DOSはアスキー

OS-9, OS-9/68000, OS-9000, MW CはMICROWARE

UCSD p-systemはカリフォルニア大学理事会

WordStar, WordMasterはWORDSTAR International

TURBO PASCAL, TURBO C, SIDEKICKはBOLAND INTERNAT

IONAL

LSI CはLSI JAPAN

HuBASICはハドソンソフト

の商標です。その他、プログラム名、CPU名は一般に各メーカーの登録商標です。本文中では"TM", "R"マークは明記していません。

本誌に掲載されたプログラムの著作権はプログラム作成者に保留されています。著作権法上、PDSと明記されたもの以外、個人で使用するほかの無断複製は禁じられています。

■広告目次

アートディンク	26
アイツ	180
アイビット電子	179
アクセス	184
エスピーエス	173
AVCフタバ電機	176
オーエーランド	177
キャスト	9
計測技研	174・175
コナミ	16・17
ザイン・ソフト	13
サザンエンタープライズ	183(上)
J&P	表3
システムサコム	10・11
SYSTEM HOUSE OH!	18
シャープ	表2・表4・1・4-8
ソフトクリエイト	181
九十九電機	19
T & E ソフト	15
T-ZONE/マイコンゾーン	182
デンキヤ	178
日コン連企画	172
パソコンプラザオクト	20・21
P & A	22・23
ビクター音楽産業	14
満開製作所	183(下)
ラオックス	24・25
ロゴシステム	12



ディスプレイ関連

カラーディスプレイテレビ



15型カラーディスプレイテレビ
CZ-602D-BK・-GY
標準価格 99,800円(税別)
(チルトスタンド同梱)

カラーディスプレイ



14型カラーディスプレイ
CZ-603D-BK・-GY
標準価格 84,800円(税別)
(チルトスタンド同梱)



15型カラーディスプレイテレビ
CZ-605D-BK・-GY
標準価格 115,000円(税別)
(スピーカー2個・チルトスタンド同梱)



14型カラーディスプレイ
CZ-604D-BK・-GY
標準価格 94,800円(税別)
(スピーカー2個・チルトスタンド同梱)

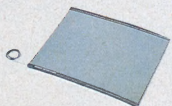


15型カラーディスプレイテレビ
CZ-613D-TN・BK・-GY
標準価格 135,000円(税別)
(スピーカー2個・チルトスタンド同梱)



21型カラーディスプレイ
CU-21HD
標準価格 148,000円(税別)
(スピーカー2個同梱)

CRTフィルター



高性能CRTフィルター
BF-68PRO
標準価格 19,800円(税別)
(14/15型用)



RGBシステムチューナー
CZ-6TU-BK・-GY
標準価格 33,100円(税別)
(リモコン付)

※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のバラレルデータ伝送を行う場合、別売のスクナ用バラレルボードCZ-6BN1標準価格29,800円(税別)で接続してください。
※2 CZ-603D/604D、CU-21HDをご使用の場合は、RGBシステムチューナーCZ-6TU(別売)が必要です。
※3 別売の信号ケーブルIO-73CX標準価格5,500円(税別)で接続して下さい。

アートツール

画像入力



カラーイメージスキャナ^{※1}
CZ-8NS1
標準価格 188,000円(税別)



スクナ用バラレルボード
CZ-6BN1
標準価格 29,800円(税別)

映像入力



カラーイメージユニット^{※2}
CZ-6VT1-BK
CZ-6VT1
標準価格 69,800円(税別)

プリンタ

カラープリンタ



24ドット
熱転写カラー漢字プリンタ
CZ-8PC3
標準価格 65,800円(税別)
(信号ケーブル同梱)



48ドット
熱転写カラー漢字プリンタ
CZ-8PC4
CZ-8PC4-GY
標準価格 99,800円(税別)
(信号ケーブル同梱)



カラービデオプリンタ
CZ-6PV1
標準価格 198,000円(税別)
(信号ケーブル同梱)

カラーイメージジェット



カラーイメージジェット^{※3}
IO-735X
標準価格 248,000円(税別)
(信号ケーブル別売)

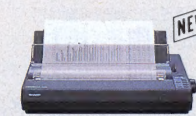
ドットプリンタ



24ピン
カラー漢字プリンタ(80桁)
CZ-8PG1
標準価格 130,000円(税別)
(信号ケーブル同梱)



24ピン
カラー漢字プリンタ(136桁)
CZ-8PG2
標準価格 160,000円(税別)
(信号ケーブル同梱)



24ピン漢字プリンタ(136桁)
CZ-8PK10
標準価格 97,800円(税別)
(信号ケーブル同梱)

ファイル

ハードディスク



ハードディスクユニット(20MB)
CZ-620H
標準価格 178,000円(税別)



増設用ハードディスク
ドライブ(40MB)
(CZ-602C/652C/603C/
653C内蔵用)
CZ-64H
標準価格 120,000円(税別)
(取付費別)

※取付に関してはシャープ
お客様ご相談窓口にてご
相談ください。

AVturbo シリーズ用 周辺機器

標準価格は税別です。

カラーディスプレイ

●21型カラーディスプレイ^{※1} CU-21HD 148,000円

映像・画像入力編集装置

●カラーイメージスキャナ CZ-8NS1 188,000円
●カラーイメージボードII CZ-8BV2 39,800円

●立体映像セット ★CZ-8BR1 29,800円
●パーソナルテロップ^{※2} CZ-8DT2 44,800円

FM音源

●ステレオタイプFM音源ボード CZ-8BS1 23,800円
スピーカー(2本1組)標準装備、ミュージックツール同梱

プリンタ

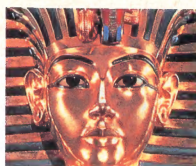
●24ピンカラー漢字プリンタ(80桁) CZ-8PG1 130,000円
●24ピンカラー漢字プリンタ(136桁) CZ-8PG2 160,000円

●24ピン漢字プリンタ(136桁) CZ-8PK10 97,800円
●24ドット熱転写カラー漢字プリンタ CZ-8PC3 65,800円
●48ドット熱転写カラー漢字プリンタ CZ-8PC4 99,800円
●48ドット熱転写カラー漢字プリンタ CZ-8PC4-GY 99,800円
●カラービデオプリンタ CZ-6PV1 198,000円
●カラーイメージジェット IO-735X 248,000円

ファイル

●ミニフロッピーディスクユニット(2HD・2D)^{※3} ★CZ-520F 118,000円

X68000をサポート。



シャープペリフェラルファミリー X68000



ボード

拡張メモリ



1MB増設RAMボード
(CZ-600C専用)
CZ-6BE1
標準価格 35,000円(税別)



1MB増設RAMボード
(CZ-601C/611C/652C/653C/662C/663C用)
CZ-6BE1B
標準価格 28,000円(税別)



2MB増設RAMボード※4
CZ-6BE2
標準価格 79,800円(税別)



4MB増設RAMボード※4
CZ-6BE4
標準価格 138,000円(税別)

インターフェイス



ユニバーサルI/Oボード
CZ-6BU1
標準価格 39,800円(税別)



GP-IBボード
CZ-6BG1
標準価格 59,800円(税別)



増設用RS-232Cボード
(2チャンネル)
CZ-6BF1
標準価格 49,800円(税別)

数値演算プロセッサ



数値演算プロセッサボード
CZ-6BP1
標準価格 79,800円(税別)

FAX



FAXボード
CZ-6BC1
標準価格 79,800円(税別)

MIDI



MIDIボード
CZ-6BM1
標準価格 26,800円(税別)

ネットワーク

モデム



モデムユニット※5
CZ-8TM2
標準価格 49,800円(税別)
(RS-232Cケーブル同梱)

RS-232Cケーブル



RS-232Cケーブル
(平行接続型)
CZ-8LM1
標準価格 7,200円(税別)



RS-232Cケーブル
(クロス接続型)
CZ-8LM2
標準価格 7,200円(税別)

LANボード



LANボード
CZ-6BL1
標準価格 268,000円(税別)
CZ-6BL2
標準価格 298,000円(税別)
※電源ユニット・ソフトウェア
(ネットワークドライブVer1.0)同梱

入力



インテリジェントコントローラ
CZ-8NJ2
標準価格 23,800円(税別)



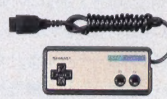
マウス・トラックボール
CZ-8NM3
標準価格 9,800円(税別)



トラックボール
CZ-8NT1
標準価格 13,800円(税別)



マウス
CZ-8NM2A
標準価格 6,800円(税別)



ジョイカード
CZ-8NJ1
標準価格 1,700円(税別)

その他

拡張スロット



拡張I/Oボックス(4スロット)
(CZ-600C/601C/611C/602C/612C/603C/613C/623C用)
CZ-6EB1-BK
CZ-6EB1
標準価格 88,000円(税別)

スピーカー



アンプ内蔵
スピーカーシステム(2本1組)
AN-S100
標準価格 36,600円(税別)

システムラック



システムラック
(CZ-600C/601C/611C/602C/612C/603C/613C/623C用)
CZ-6SD1
標準価格 44,800円(税別)

※4 ご使用に際しては、あらかじめ別売の1MB増設RAMボードCZ-6BE1 標準価格35,000円(税別・CZ-600C用)、CZ-6BE1B 標準価格28,000円(税別・CZ-601C、CZ-611C、652C、653C、662C、663C用)を増設してください。
※5 モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。

●ミニフロッピーディスクユニット(2D)	★CZ-502F	99,800円
●ミニフロッピーディスクユニット(2D・1ドライブ)	CZ-503F	49,800円
●増設用ミニフロッピーディスクドライブ(2D)※4	CZ-53F-BK	19,800円

拡張ボード・その他

●モデムユニット(300/1200ボー)	CZ-8TM2	49,800円
●320KB外部メモリ	CZ-8BE2	29,800円
●RS-232C・マウスボード※5	CZ-8BM2	19,800円
●フロッピーディスクインターフェイス※6	CZ-8BF1	14,800円

●JIS第1水準漢字ROM※7	CZ-8BK2	19,800円
●RS-232C用ケーブル(平行接続型)	CZ-8LM1	7,200円
●RS-232C用ケーブル(クロス接続型)	CZ-8LM2	7,200円
●拡張I/Oボックス	CZ-8EB3	33,800円
●RFコンバータ※8	AN-58C	2,980円
●インテリジェントコントローラ	CZ-8NJ2	23,800円
●マウス・トラックボール	CZ-8NM3	9,800円
●マウス	CZ-8NM2A	6,800円
●トラックボール	CZ-8NT1	13,800円

●ジョイカード	CZ-8NJ1	1,700円
●チルトスタンド	CZ-6ST1-E・B	5,800円
●高性能CRTフィルタ※9	BF-68PRO	19,800円
●スキャナ用パラレルボード※10	CZ-8BN1	27,800円
●品番中の一表示は、B<ブラック>・E<オフィスグレー>を示します。※1 X1ターボシリーズ用 ※2 CZ-862Cには接続できません。※3 X1ターボシリーズ用 ※4 CZ-830C用 ※5 X1シリーズ用 ※6 CZ-850CでCZ-520Fを使用する場合に必要 ※7 CZ-800C、801C、802C、803C、811C、820C用 ※8 CZ-820C、822C、830C用 ※9 J4/J5型用 ※10 CZ-8NS1用 ●接続等の説明につきましては、周辺機器総合カタログをご参照ください。		

★印の商品は在庫僅少です。

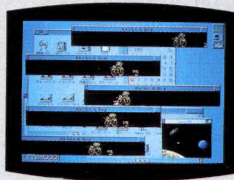
本広告に掲載しております商品および役務の価格には消費税は含まれておりませんので、ご購入の際、消費税額をお支払い下さい。

SHARP

"アート"と呼べる高水準のソフトウェアが

(次代のインテリジェンス、
ウィンドウ環境をあなたのX68000で。)

ユーザー本位の操作環境を提供するフル画面マルチウィンドウタイプの美しいデスクトップ(テキスト面/単色4階調+カラー4色、グラフィック面/カラー65,536色中16色)、イベントドリブン型マルチタスク処理により複数の作業を同時に処理できる疑似マルチタスクや入出力装置の設定が簡単に行える多機能コントロールパネルを搭載した本格ウィンドウシステムです。従来のビジュアルシェルのとは異なり、今後のアプリケーションソフトが統一された操作環境で実行できるようになります。



SX-WINDOW ver1.0

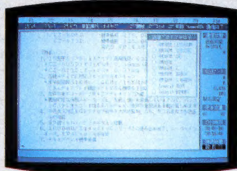
CZ-259SS 6月発売予定



(ソースコードデバッグをはじめ、
各種開発ツールを強化。
バージョンアップされたCコンパイラ。)

Communication PRO-68Kのバージョンアップ版です。300BPSから19,200BPSまでの通信速度に対応し、パソコン同士の接続や各種データベースの漢字端末に、またホストコンピュータとのデータ通信に利用できます。さらにMNPモデムへの対応で、ハードフロー制御(GTS/RTS)をサポート。その他、高速逆スクロール機能、オートログイン/オートパイロットが可能な自動実行機能、コンカレント機能も装備。行入力機能やスクリーンエディタなど豊富な編集機能も魅力です。

また、バイナリファイルを転送するプロトコルとしてXmodem(128/SUM, 128/CRC, 1K)、Ymodem(G, BATCH, G-BATCH)、Transit2(TEXT, BINARY)プロトコルもサポートしています。



CZ-257CS 6月発売予定

Communication PRO-68K

Cのソースレベルでデバッグできるソースコードデバッグを搭載したほか、各種開発ツールを強化した総合開発ツールです。また、ライブラリはHuman 68k ver2.0の拡張DOSコールもサポートしているなど、よりX68000のハードウェアを活かせる豊富なライブラリ(約800種)となっています。C言語の最も基本的なK&R仕様を充たし、ANSI規格案に準拠した最新のCコンパイラです。「BASIC-Cコンバータ」、「アセンブラ」、「リンク」、「デバッグ」、「ソースコードデバッグ」、「アーカイバ」、「ライブラリアン」、「コンバータ」などのツールが装備されています。



CZ-245LS 7月発売予定

C compiler PRO-68K

X68000をサポート。



シャープオリジナルソフトウェア
X68000

ビジネスツール

Hyperword

■CZ-251BS 標準価格39,800円(税別)

X68000の優れたグラフィック環境を活用し効率的に文書を作成するためのインテリジェントワープロです。アイデアプロセッサ機能、ハイパーテキスト機能などをサポート。データの整理やプレゼンテーションツールなど幅広い用途に利用できます。



TOP給与計算エキスパート

■CZ-228BS 標準価格200,000円(税別)

給与計算から明細発行までを、リアルイメージ入力により自動的に、素早く処理することができます。

TOP財務会計

■CZ-227BS 標準価格200,000円(税別)

会計エキスパートシステムとデータベースを搭載し、機能と操作性を両立させた財務会計ソフト。

CYBERNOTE PRO-60K

■CZ-243BS 標準価格19,800円(税別)

プライベートなデータやビジネスデータを簡単な操作で管理・運営できるパーソナルデータベースです。リフィル、タックシール、ハガキなどへの印字もOK。シャープ電子手帳とのデータ交換可能(別売の通信ケーブルCE-200Lが必要)。



CARD PRO-60K

■CZ-226BS 標準価格29,800円(税別)

自由なレイアウト画面で入力できるワープロ機能を装備したカード型リレーショナルデータベース。

CARD PRO-68K用システム手帳リフィル集

■CZ-241BS 標準価格9,800円(税別)

CARD PRO-68K用活用フォーム集

■CZ-242BS 標準価格9,800円(税別)

Stationery PRO-60K

■CZ-240BS 標準価格14,800円(税別)

他のソフトを起動する前に、このStationery PRO-68Kを一度起動するだけで、他のソフトを実行中に「スケジュール」「住所録」など多彩な機能をワンタッチで使用できます。シャープ電子手帳とのデータ送受信も実現。(別売の通信ケーブルCE-200Lが必要)。



DATA PRO-60K

■CZ-220BS 標準価格58,000円(税別)

入力の手間を軽減するヒストリー機能を装備した、コマンド型リレーショナルデータベースです。

BUSINESS PRO-60K

■CZ-212BS 標準価格68,000円(税別)

スプレッドシート(表計算)、データベース、グラフ作成機能を一体化させた統合ビジネスツールです。

グラフィックライブラリ VOL.1

■CZ-235GS 標準価格8,800円(税別)

著中見舞用を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。

グラフィックライブラリ VOL.2

■CZ-236GS 標準価格8,800円(税別)

年賀状を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。



アートツール

NEW PrintShop PRO-60K

■CZ-221HS 標準価格19,800円(税別)

オリジナリティあふれるはがき等、簡単に作成、印刷できるホームプロダクティビティツール。ほとんどの処理をアイコンで表示しマウスで選ぶフレンドリーオペレーション。



Musicstudio PRO-60K ver.1.1

■CZ-252MS 標準価格28,800円(税別)

24トラック対応MIDIマルチレコーディングソフトMusicstudio PRO-68Kがバージョンアップしました。従来の機能に加え、小節間のコピー及びデリートや、MIDIインプットモニターなど、数々の機能を追加・改良。さらに使いやすくなりました。
※MIDIボード(CZ-6BM1)が必要です。



MUSIC PRO-60K [MIDI]

■CZ-247MS 標準価格28,800円(税別)

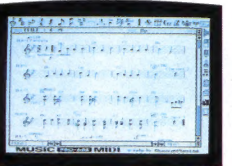
MIDI対応自動伴奏機能をサポート、簡単な楽譜入力でMIDI演奏が楽しめます。

※MIDIボード(CZ-6BM1)が必要です。

ソングライブラリ<101曲集>

■CZ-248MS 標準価格8,800円(税別)

鑑賞用と音楽データ加工作成からなるライブラリです。



Sampling PRO-60K

■CZ-215MS 標準価格17,800円(税別)

AD PCM機能を活かす高機能サンプリングエディタ。多彩なEDITORを装備、サンプリング音のデータはBASICでも活用できます。

SOUND PRO-60K

■CZ-214MS 標準価格15,800円(税別)

スタジオのコンソールパネルを操作する感覚でFM音源による音創りが楽しめるサウンドエディタ。

MUSIC PRO-60K

■CZ-213MS 標準価格18,800円(税別)

最大8パートのスコア(総譜)が書け、内蔵のFM音源で演奏できる楽譜ワープロ・演奏用ツール。



シューティングゲーム
「ツインビー」
■CZ-217AS 標準価格7,800円(税別)
© KONAMI 1988



シューティングゲーム
「沙羅曼蛇」
■CZ-218AS 標準価格7,800円(税別)
© KONAMI 1989



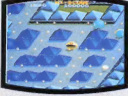
ブロックゲーム
「アルカノイド」
■CZ-222AS 標準価格7,800円(税別)
© TAITO CORP 1987



ドライブゲーム
「フルスロットル」
■CZ-231AS 標準価格8,800円(税別)
© TAITO CORP 1988



スポーツゲーム
「熱血高校ドッジボール部」
■CZ-232AS 標準価格7,800円(税別)
© TECHNOS JAPAN CORP 1988



アクションゲーム
「バックマニア」
■CZ-233AS 標準価格7,800円(税別)
© NAMCO



アクションゲーム
「ニューゼalandストーリー」
■CZ-230AS 標準価格8,800円(税別)
© TAITO CORP 1989



スポーツゲーム
「V'BALL」
■CZ-246AS 標準価格7,900円(税別)
© TECHNOS JAPAN CORP 1989



バイクレーシングゲーム
「スーパーハンガオン」
■CZ-238AS 標準価格8,800円(税別)
© SEGA 1987



ジェットヘリシミュレーションゲーム
「サンダーブレード」
■CZ-239AS 標準価格9,500円(税別)
© SEGA 1987



アクションゲーム
「ダウンタウン熱血物語」
■CZ-254AS 標準価格8,800円(税別)
© TECHNOS JAPAN CORP 1989

開発ツール

OS-9/X68000

■CZ-219SS 標準価格29,800円(税別)

OS-9のもつマルチタスク機能、リアルタイム機能を活かした使いやすい機能的なOS環境を提供。これまでのデータ資産も活かれます。
※OS-9はマイクロウェア社の登録商標です。

Human68k ver2.0

■CZ-244SS 標準価格9,800円(税別)

THE福袋V2.0

■CZ-224LS 標準価格9,980円(税別)

AI-68K (Staff LSP/OPS PRO-68K)

■CZ-234LS 標準価格188,000円(税別)

サウンドツール

本広告に掲載しております商品および役務の価格には消費税は含まれておりませんので、ご購入の際、消費税額をお支払い下さい。

EXEクラブが待っている。

●X68000を手にしたら何はともあれ「EXEクラブ」へ。トレンド的な仲間が熱烈歓迎。同梱のハガキをすぐに返送してください。折り返しX68000 オリジナルの会員証電卓をお手元に……。またEXEクラブニュースや新作ソフト、周辺機器などX68000の最新情報も随時ご送付。各種イベント、フェアへのご招待も。もちろん会費は無料です。

●EXEリーダーズグッズ、プレゼント実施中(EXE会員のみ)
あなたの知人、友人やX68000に興味をお持ちの方をぜひご紹介ください。ご紹介くださった方が期間中(平成2年9月30日まで)X68000を購入されますと、あなたにすてきなリーダーズグッズをプレゼント。詳しくはお近くのEXEショップまでお問い合わせください。

NEW X68000、新作ソフト、面白イベント…… まるごと見・体・験フェア。

●ご存じ「見体験フェア」、今回のテーマはやはりニューX68000。SUPER-HD/EXPERTII/PROIIの魅力を直にご体験ください。業界注目のSX-WINDOWもまるかじり。MIDIシステムの「トレンドテーマコーナー」、新作ソフト体験コーナー、100インチ液晶プロジェクションによる大迫力のゲームたちに出会えるかも。人気のX68オリジナルグッズも展示即売!! お近くの会場へぜひお立ち寄りください。

NEWミュージックトレンド「MIDI」体験!! 必聴、必見「音遊サウンドライブ」。

●X68000が創造するパソコントレンド、C.G.の次はMIDI。デジタル世代を自負するならこれに乗り遅れてはいけません。これは自明の理。90年代の音楽シーンはMIDIなくしては語れない。楽器のプロもパソコンのプロも、どちらも一度は手を染めてみたいのがこのフィールドだ。そして、音楽さえ好きであれば、どちらも苦手な人も即エンターティナーになれるところがウレシイ。自分の好きな曲を、好きな楽器で、好みのリズムで演奏できる、いま注目度一番のニュートrend。プロのキーボード奏者による本格的なMIDIライブコンサートで、キミの感性をとがらせよう。今、全国各地で開催中!

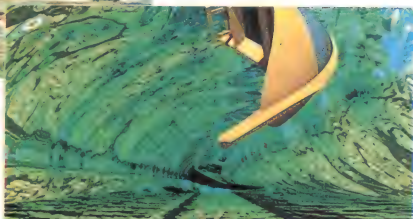
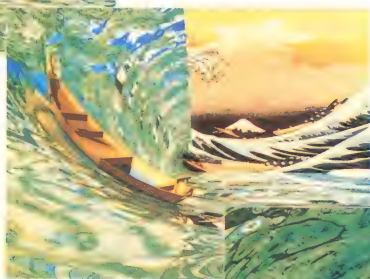
音遊サウンドライブ
in Summer



彼女が「EXEクラブ」に入会したら……。

●見体験フェア/音遊サウンドライブ 開催日程

開催月日	開催地区	開催場所	お問い合わせTEL
6/23(土)	神奈川	ICコスモランドあざみ野店フェア	(045)903-1010
6/23(土)・24(日)	東海	ジャスコ豊田店フェア	(0565)35-1761
6/23(土)・24(日)	東京	九十九電機7号店フェア	(03)253-4199
6/24(日)	東京	T-ZONEフェア	(03)257-2650
6/30(土)・7/1(日)	中部	コムライン豊田店フェア	(0565)35-1150
6/30(土)・7/1(日)	東海	ジャスコ岡崎店フェア	(0564)23-4960
7/1(日)	神奈川	ウェーブ・アイ大和店オープンフェア	(0466)43-1711(湘南台店)
7/7(土)・8(日)	東北	デンコードー八戸本店 OAフェア	(0178)44-4111
7/14(土)・15(日)	北海道	シャープグランドフェア イン 釧路	(0154)22-9107
7/29(日)	北陸	シャープ見体験フェア イン 富山	(0762)49-1181



▶3次元レイトレーシングソフト◀

C-TRACE98 **Ver.3.0**
(PC-9801シリーズ対応) ¥98,000

C-TRACE68 **Ver.3.0**
(X68000シリーズ対応) ¥98,000

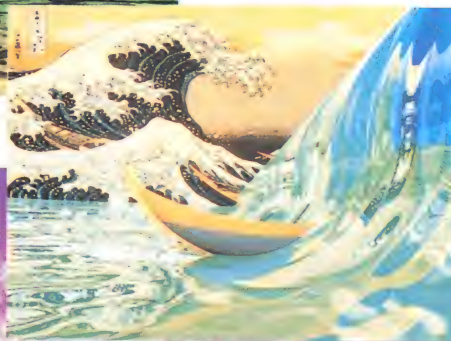


バージョンアップ受付中。

誕生!

〈NEW VERSION〉
Ver.3.0

ボクセル分割による高速化。
アニメーション機能の充実。
マッピング環境の充実。

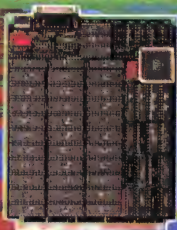


98ユーザーに朗報!!

Ver.3.0 誕生を記念して、
当社に直接お申し込みの方

先着100名様に、も

フルカラー
フレーム
バッファ
がついてくる。



——アニメーション用ローダー——

68用

アニメーションツールボックス「うごくZO」ARGOより発売
価格 ¥7,800 (税込み) C-TRACEクラブ会員価格 ¥5,800 (税込み)
問い合わせ TEL.03-5996-4459

98用

フルカラーフレームバッファ「画像」の付属ソフト「anime」が使用できます

C-TRACE TOWNS

¥68,000

C-TRACE NEWS Ver.3.0 (SONY) ¥530,000 新発売

★C-TRACE 98TP

¥670,000

★C-TRACE 68TP

¥670,000

表示価格に消費税は含まれません。

★の製品は店頭販売いたしておりません。直接当社までお申し込みください。

Cast

株式会社キャスト

●お問い合わせ先●

〒158 東京都世田谷区等々力2-1-13
TEL.03-705-1065 FAX.03-705-5224

闇の血族

THE PREDESTINED HOMICIDES #1

美少女名探偵 魅由の繰り広げる
ミステリアスアニメーションアドベンチャー第1弾!!

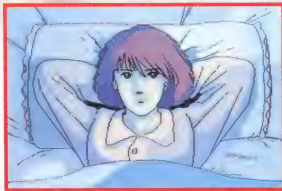
艶やかなファッション界を襲う奇怪な連続殺人事件。
南米の血に隠された秘密とは?
そして魅由を待ち受ける血族の宿命は?

NOVELWARE

あたし、魅由。

新宿にあるデザイン・スタジオの、新人A・D（アパレル・デザイナー）。……なんだけどあたしの持つてた妙な「力」みたいなモノ——人の心が判っちゃったり、変にカンが良かったり——のせいで、周りからは「名探偵魅由」なんて呼ばれて、よく相談事を持ち込まれたりしている。で、そんなある日、友達のモデルが、突然、殺されてしまった。

そして、あたしの親友だった唯も……/
これって……ひょっとして連続殺人事件ってヤツ?!



新発売!!

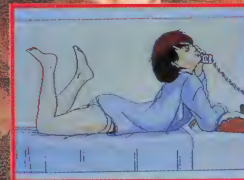


X68000対応 5"-2HD

●ローランド社MT32完全対応
MIDIインターフェイスボードC-2-6BML
又は、SACOM製SX-68Mが必要です。

(初期のMT-32では、正常に演奏できません)

標準価格 8,800円



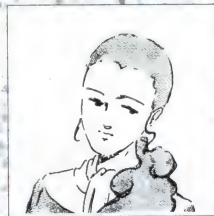
伊澤 魅由
(いざわ みゆ)
19才

誕生日: 7月16日
身長: 168cm
体重: 59kg



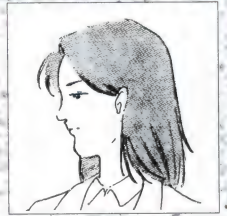
姫野 里沙
(ひめの りさ)
18才

誕生日: 4月2日
身長: 163cm
体重: 45kg



雪原 リーン
(ゆきはら リーン)
20才

誕生日: 2月10日
身長: 170cm
体重: 53kg



高校生の時、デザイナーの泉麗子に見込まれ、学生生活を営む傍ら麗子のデザインスタジオ（専門学校）に通い始める。そこで小品の手伝いなどをしながら、デザイナーとして本格的に勉強を開始。2年間の研修期間を終え、高校卒業と同時に麗子の強力な推薦で、現在所属している〈スタジオYo〉に入った。

趣味: ジョギング、ティー・タイム 好きな食べ物: チーズケーキ 嫌いな食べ物: プロッコリー 好きな音楽: なんでも大好き、現在はディズニー音楽に夢中とか
家族構成: 父、母、妹

〈スタジオYo〉の専属モデル。ファッションショー、雑誌モデルを専門としている。平凡な可愛さがウリで、生活の中で“Yo（自己性）”をファッションナブルに演出する——といった〈スタジオYo〉のメイン・コンセプトから考えれば、最もYoらしいモデルと云えるかも知れない。

趣味: 読書、推理ものが好きとかで、本格的シャーロックアンでもある。好きな食べ物: 果物、全部 嫌いな食べ物: ピーマンだけが、苦手 好きな音楽: 主に、ニューエイジ・ミュージックをよく聞く。特に好き嫌いなし。
家族構成: 父、母、兄

〈スタジオYo〉の付属学校、「矢萩デザイナーズ・スタジオ」の卒業生。研修期間中「Yoプロデュース」でスタイリスト補助のアルバイトをしていた。現在では、Yoでファッションショーを中心とした若手スタイリストとして活躍中。

趣味: 読書、ジョギング 好きな食べ物: 特になし 嫌いな食べ物: コンニャク 好きな音楽: クラシック 家族構成: 父、母、弟

ゲームセンターを賑わした大人気シューティングゲーム
「ジェミニウイング」が、キミのX68Kで今、蘇る!!

ジェミニウイング

Gemini Wing™

7月上旬発売予定

アーケード版
ジェミニウイング
待望の移植を実現!

MIDI対応

幾千の流星が降りそそいだ年、世界は蟲に覆われていた。人々は孤立し、街は滅び、植物に埋め尽くされた。蟲たちはさらに勢いを増し、残された僅かな地さえも蝕んでゆく。そして、ついに最高機密指令第307号、コード名 ジェミニウイングは発動された……!

◆特徴◆ ●二人同時プレイ可能 ●5"2HD 2枚組

●MIDI対応(※)

●FM音源、ADPCM対応

X68000

対応楽器 ローランド MT-32

●ジョイスティック対応

全シリーズ対応

CM-32L CM-64

●縦横画面モード対応

(※)対応機種ごとに、それぞれ違ったBGMをお楽しみいただけます。標準価格8,800円

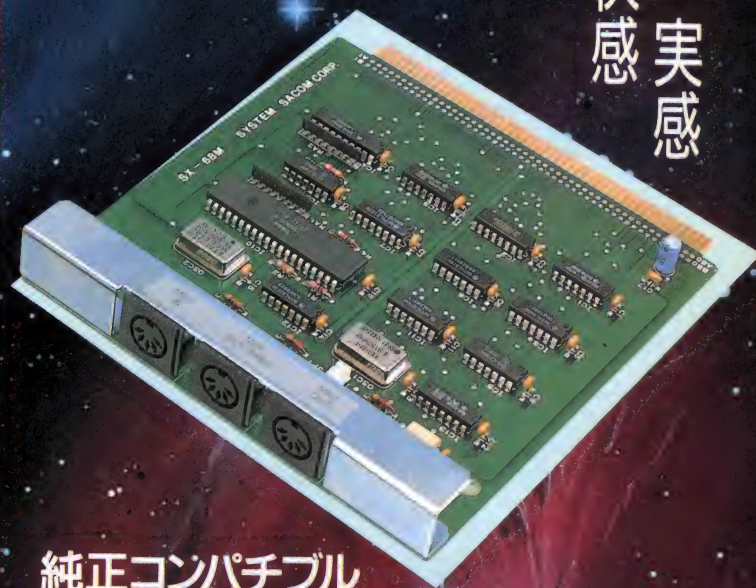
(注)初期のMT-32では正常に演奏できません。

△68000 SERIES

MIDI INTERFACE BOARD

SX-68M

体感 快感 実感



純正コンパチブル

「SX-68M」は、純正品との互換性を保ちつつ、さらに、お求めやすい低価格におさえた、X68000シリーズ専用MIDIインターフェースボードです。特徴としては、ボード本体に直接MIDI規格のDINコネクタを装着することによって、中間に変換ケーブルを使用する必要がなくなりました。また、クロック部に安定度の高いオシレーターを採用することにより、さらに信頼度の高いものとなっております。もちろん、従来のMIDIボードをサポートするソフトウェアはそのままお使いになれます。SX-68Mで、あなたもすばらしいMIDIの世界を体験してください。

(※)本ボードは、TAPE SYNC端子を装備していないため、その機能をサポートするソフトは、ご使用いただけません。また、本ボードは、2枚同時装着できませんので、ご注意ください。

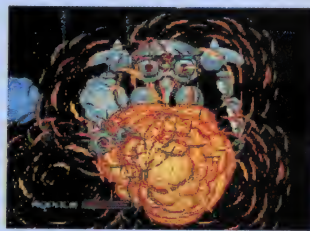
SX-68M仕様

品 名	MIDIインターフェースボード
規 格	MIDI規格 1.0準拠
コントロール LSI	日本楽器(YAMAHA) YM3802
MIDI端子	MIDI OUT 2端子 MIDI IN 1端子 MIDI OUT 1端子 MIDI THRU 1端子 MIDI IN 1端子
電 源	+5V 170mA(本体より供給)
外形寸法	150mm(W)×167mm(D)×23mm(H)
重 量	約160g
標準価格	¥19,800

対応ソフト紹介



■38万キロの虚空



■メタルサイト

東芝EMIより『38万キロの虚空』CD

7月11日 新発売!!

税抜価格 2,184円

税込価格 2,250円

SACOM

株式会社 システム サコム

ソフトウェア部 TEL.03(635)7609

〒130 東京都墨田区両国4-38-16
両国桜井ビル4F

ハードウェア部 TEL.03(635)5145

TEL. 03-635-7609

※標準価格には消費税は含まれておりません。

68000

本格的ファイルマネージングソフトウェア

**業界の新星、ロゴスシステムが
ユーザーの希望を1つの形にしました。
これは必要だとか便利じゃない、快感だ!**

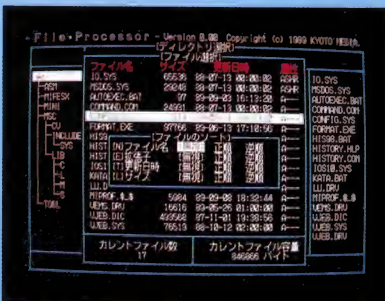
全国有名パソコンショップでお求め下さい。
電話1本での通信販売も受付いたしております。

THE FILE PROFESSORの実力

ディスクのバックアップ、ディスクのエディット、ディスクの初期化、ディスクの比較、ディスクの検査、ディスクの情報、FATのエディット、ファイルの検索、ディレクトリのコピー、ディレクトリの削除、ボリュームラベルの設定、ディレクトリの作成、ディレクトリ構造の再読み込み、ディレクトリ構造の印刷、ディレクトリ名の変更、ディレクトリ内容のソート、削除ファイルの復元、ファイル属性の変更、ファイルのコピー/移動、ファイルの削除、ファイルのエディット、ファイルの配置情報、ファイル一覧の印刷、ファイル名の変更、ファイルのソート、ファイル更新日時の変更、ファイルの表示、ファイルの発行、カレンダー、ハードディスクの直搬エディット、システム情報の表示、コマンドシェル、現在時刻の変更。

**メニュー選択方式を実現!!
初心者でも簡単に使える**

(画面写真は、98用を開発中のものです)



ロゴスシステム

このソフトはロゴスシステムのデビュー作です。でも、だからといってなめてもらっちゃあ困ります。私達は、いろいろなソフトを作りました。そのどれもが他社から発売されてきました。出来る事ならば自分達で発売したい/その願いがやっとかないました。

好評発売中!

ロゴスシステム

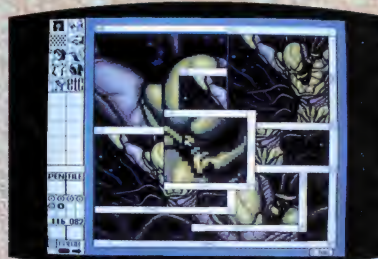
〒615 京都市右京区西院上今田町17-1 L&Pビル4F
TEL (075) 812-6383 FAX (075) 822-6915

定価 **28,000円**

The File Professor

X68000ユーザーのクリエイティブマインドに火をつける新感覚のグラフィックツール。これまでのエディタ概念を払拭し、作品に挑むうえで必要不可欠なグラフィックキャラクタ・背景作成のすべてを備えたトータルツールです。ゲームデザインをはじめとしたオリジナルコンピュータアートが驚くほど自由に多彩に描けます。今回はこの「G=ツール」の解析シリーズを展開するにあたって、その驚異の機能概要とシステム環境をご紹介します。

「GR EDITモード」と「BG EDITモード」を装備。



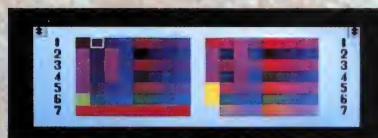
グラフィックやスプライトのキャラクタの作成を目的とした「GR EDITモード」、背景やそのキャラクタ作成を目的とした「BG EDITモード」を装備、C.G.を作品として仕上げるうえで必要なエディットツールをすべてサポート。そして、その機能のひとつひとつに新しい感覚が取り込まれています。

GR EDITモード ●最大12枚の描画ウィンドウが開けるマルチウィンドウシステム ●自分なりのアイコンメニューが作れる便利なユーザーアイコンシステム ●マウス機能定義システム ●高速メニューウィンドウ処理

BG EDITモード ●最高80×80ドットのキャラクタ作成が可能 ●キャラクタチェックができるアニメーション機能 ●1~16倍の拡大画面での描画が可能 ●強力な外部関数を装備 ●全画面スクロールが可能な背景作成



またジョブ画面のメインアイコンには、ファイル/パレット/タイル/ペン/描画動作/編集/文字/マスクなどをわかりやすく配置。それぞれポップアップメニューで仕事を選べます。さらに好きなアイコンを自由に置けるユーザーアイコンエリアを装備、使いやすいシステム環境を提供します。



「G=ツール」新登場。

7月中旬発売予定
¥28,000

もう逃げられない!

世界サイズの興奮!



世界中で数々の金字塔を打ち立てたリアルタイムRPG「ダンジョン・マスター」の興奮は本物だった。
3Dグラフィックスに展開される奥の深い迷路、数々のトリック、パーティーを突然襲って来る不気味なモンスター、
組合せと熟練度によって決定される魔法、それぞれの武器によって異なる攻撃方法、
そして何よりもプレイヤーの思考、行動にリアルタイムで反応する見事なゲーム・システム……

これが本当のRPGだ!



ダンジョン・マスター

好評発売中

■X68000
マウス対応

■PC-9801VM21/11, VX, RX, RS, RA ■PC-98DO

■PC-9801UV21/11, UX, CV, EX, ES 要バス・マウス/アナログRGB対応

各 ¥9,800 (税別)

Produced by FTL Games Copyright © 1987, 1990 Software Heaven, Inc. Copyright © 1990 VICTOR MUSICAL INDUSTRIES, INC.

■発売 ビクター音楽産業株式会社

通信販売

当社の商品をお近くのパソコンショップでお買い求めになれない場合、商品名、機種名、住所、氏名、電話番号を明記のうえ、下記住所まで定価プラス3%消費税分を現金書留にてお申し込み下さい。(送料無料) 〒151 東京都渋谷区千駄ヶ谷2-8-16 ビクター音楽産業株(通信販売係)

「ダンジョン・マスター」
オリジナルTシャツ(Mサイズ)
¥1,800(送料込み)にて限定発売!



ご希望のかたは現金書留にて
下記通帳係までお申し込み下さい

夢中宣言'90

by T&E SOFT

'90年後半押寄せの新波。

X68000ヒッグ3タイトル連続リリース!!

RUNE WORTH

第1弾 黒衣の貴公子

UNDER NO LINE

第2弾 幻獣鬼

NEW 3D GOLF SIMULATION

第3弾 遙かなるオーガスタ

T&EのBIGウェーブにご期待下さい。

△68000



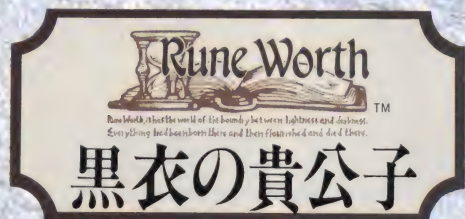
△68000

シリーズ

第1弾!

7月13日(金)発売決定

X68000 5"2HD ¥8,800 ※表示価格に消費税は含みません。



●PC-9801VM、UVシリーズ PC-286、386シリーズ、NOTE対応

5"2HD/3.5"2HD 2枚組 ●サウンドボード対応 ●ジョイスティック対応

※VM UVはRAM容量640Kバイト以上必要です。

※PC-9801/E/F/M/VF/U2/XAでは、ドライブ、RAM等の増設の設何にかかわらず、作動いたしません。また1ドライブのみ搭載の

PC-286L/LFおよびUPC-286NOTE Executiveでは、ドライブを増設しても作動いたしません。

※高解像度(640×400ドット)カラーディスプレイをお使いください。液晶ディスプレイにも対応しています。

●PC-8801SRシリーズ・VA、98DO対応 5"2D 5枚組

●サウンドボードII完全対応、ADPCMをフルサポート ●ジョイスティック対応(98DOを除く)

●NEC純正128KRAMボード、I/Oデータ機器製RAMボードに対応したキャッシュドライバ搭載

●MSX 2/MSX 2+ (RAM64K以上) 3.5"2DD 3枚組

●MSX-MUSIC (FM音源) 対応 ●ジョイスティック対応

標準 各¥8,800

表示価格に消費税は含みません。

MSX is a registered trademark of ASCII Corp.

© 1990 T&E SOFT

まだまだ続くBIGウェーブ、お楽しみに!



No Copy

このマークは
本誌コピー
禁止マークです

T&E SOFT

企画・開発・製造・販売 株式会社ティーアンドイーソフト

〒465 名古屋市中東区豊が丘1810番地 PHONE:052-773-7770

●T&Eの最新情報がわかるテレフォンサービス 名古屋(052)776-8500

世にも楽しいシューティングパズル

クオースTM

QUARTH

©KONAMI 1990

X68000版
7月6日 発売
6,800円 (税別)

MSX2版 好評発売中 5,800円 (税別)

PC-9801版 近日発売予定

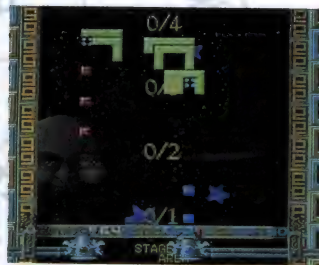
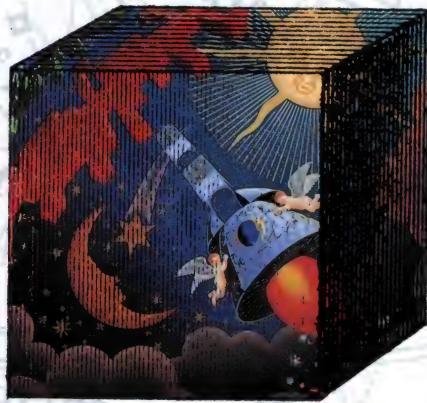
落ちて来るブロックを四角にして消してゆきます。一度にたくさん消すと効率的で得点も大幅アップ。下のラインまで来るとゲームオーバーです。





歴史に残る 前人未踏の 四角い宇宙

だれもが夢中になれるゲームを創りたい。
シンプルでいて奥が深い。だれでも気軽に遊べて、いつまでも飽きない。そんなピュアな、ほんとうの意味でのゲームがしたいと思うことがある。ゲームに対する熱い想いをもう一度じっくりと見つめて今、コナミが新たに発進する、樂園ゲームプロジェクト「クオース」。シューティングの楽しさと、パズルの思考性がマッチングした、すでにゲームセンターでは爆発人気の極楽行き超ソフトだ。ほら、もう引力がココロをズルズルと吸いこんでいる。君も、友も、父も、母も、老若男女を巻きこんで、樂園へ行こう。



協力2人で、仲直りもできる、ますます熱中の親切設計。



対戦2Pは、敵と相手の両方と戦う恐怖のケンカバトルだ。



アイテムブロックが出るラッキー。

エメラルド伝説

定価各7,800円(税別)

遙か昔、いや遠い未来かも知れない。

銀河を超えた宇宙の果てに美しき惑星「セラ」があった。惑星セラはフォースの力に守られ、それはシーズと呼ばれる5つの結晶体から生じていた。

結晶体は遠い昔に滅びたと伝えられる水の惑星「テラ」に存在したエメラルドという石の中に封印されていた。

ある時、伝説の破壊神「ソドム」が、セラに出現した。時空は歪み、生命体、物質、そしてシーズまでもが混沌の渦の中で漆黒のベールに包まれていった。

ある時、一人の少女がとある廃虚の地下室で目覚めた。その時、物語ははじまる……



RPGの世界が新しい広がりをもてる

**PC-98/88版
絶賛発売中!!**

■△▽68000版7月上旬発売!!

■**PC9801版**(5インチ、3.5インチ2HD)

(VM以降対応、要640K、LT、XL、XAシリーズ、U2を除く)

※ご注意……98版ではアナログディスプレイ、16色ボードが必要です。

■**PC8801版**(5インチ2D)

(SR以降対応版、VAシリーズ対応)

※ご注意……アナログディスプレイが必要です。

通信販売をご希望の方は…

電話にてお申し込み下さい。

着払いにてお届け致します。

●現金書留の場合………

商品名、機種、メディアを明記の上
右記迄お送り下さい。

宅配便にて急送致します。

(送料弊社負担)

SYSTEM HOUSE OH!

株式会社 システム ハウス オー
〒607 京都市山科区音羽西林町2
☎(075)502-2972

夏ツクモ! サバーゲン

商品代金2万円以上送料無料!!

※表示価格に消費税は含まれておりません。

商品のお申し込みは通販受注センター

ツクモ
オリジナル

0120-377-999

商品のお問い合わせは各店、又は
通信販売部 ☎03(251)9911へ



今、注目を集めるパソコンミュージック“MIDI”をテーマのライブショーをご案内します。

開催日 6/23(土)・24(日)午後3回(PM12:30/1:30/2:30) 場所 九十九電機7号店店頭
キーボードが弾ける人も弾けない人もX68000でMIDIを楽しくMusic、Musicstudio & Mu-1の使い方も……
わかり易く解説しますので是非お立ち寄り下さい。

ナイターセール!

ニューセンター

ツクモ7号店、N・C店は、この夏夜遅いお客様の為に6/22(金)〜7/28(金)の全・土は夜8:00迄営業 / ごゆっくりお買物ができます!

(N・C店は土のみ夜8:00迄営業)

6/20(水)〜8/20(月)迄は、夏ツクモ! サバーゲンとして5,000円以上のお買い物のお客様にドリンクをプレゼントさせていただきます。
ツクモの夏は涼しいぞ!

夏ツクモ! サバーゲン

目玉品

電話セット
38%OFF

X68000EXPERTセット

CZ-602C ¥356,000
GD-V140 ¥128,000
(14インチDP0.31mm回転台付ノンフレア)
(15〜34kHzオート)
合計定価 ¥484,000

ツクモ特価

¥298,000

(消費税別 ¥8,940)

クレジット例(税込・24回払)

初回 ¥14,880 + 月々 ¥14,700 × 23回

NEW



今のハードディスクじゃ
容量が足りない!
と考えている方へ

ハードディスク

シャープより595MBの
ディスクユニット発売!
光磁気ディスクユニット
CZ-6M01 予約受付中!
SCSIボード
CZ-6BS1 予約受付中!

アイテックハードディスク

●IT X640 定価 ¥158,000
特価 ¥89,800
●IT X680 定価 ¥198,000
特価 ¥118,000

PROII

CZ653C 定価 ¥285,000

CZ663C 定価 ¥395,000

●次世代のインテリジェンス、SX-WINDOW搭載 ● 知的ニュースタンダードフォーム ● BIOSの改良によりハイスピード処理を実現 ● 3Mバイトの大容量メモリを標準装備 ● 拡張I/Oポート4スロット標準装備

EXPERTII

CZ603C 定価 ¥338,000

CZ613C 定価 ¥448,000

●次世代のインテリジェンス、SX-WINDOW搭載 ● チェンカラーのクラティブラック ● 80MBハードディスク搭載 ● 世界標準SCSIインターフェース標準装備 ● BIOSの改良によりハイスピード処理を実現 ● 3Mバイトの大容量メモリを標準装備

SUPER HD

CZ623C 定価 ¥498,000

●次世代のインテリジェンス、SX-WINDOW搭載 ● 「チタン」カラーのクラティブラック ● 80MBハードディスク搭載 ● 世界標準SCSIインターフェース標準装備 ● BIOSの改良によりハイスピード処理を実現 ● 3Mバイトの大容量メモリを標準装備

新作ソフト紹介

日本語ワープロ

Hyper Word(CZ-251BS)

特価 ¥33,830

ウィンドウシステム

SX-WINDOW(CZ-259SS)

予約受付中!

Software tools

(GRAPHIC TOOLS)

マジックパレット 特価 ¥16,830

Z's STAFF PRO-68K 特価 ¥49,300

サイクロンExpress α88 特価 ¥83,300

デジタルクラブ 特価 ¥33,800

(電子手帳ソフト)

CYBERNOTE PRO-68K 特価 ¥16,830

Stationery PRO-68K 特価 ¥12,580

※通信ケーブル CE-300L 特価 ¥2,520

電子手帳

PA-8600 特価 ¥24,800

PA-6500 特価 ¥9,800

プリンター

CZ-8PG1 定価 ¥130,000

CZ-8PG2 定価 ¥160,000

CZ-8PC3 定価 ¥65,800

CZ-8PC4 定価 ¥99,800

IO-735X 定価 ¥248,000

※特価はお電話にてお問い合わせ下さい!

通信モデム&ソフト

アイワ

PV-A24MNP5(モデム) 45%

定価 ¥54,800 特価 ¥29,800 off

た〜みのる2(ソフト)

ツクモ特価 ¥15,000

TSUKUMO NET

新規会員募集!! この度、

X68000PROのホストシステムへ

移行し、3回線までサポートしました。

入会希望の方は7号店荒井まで!

回線番号

☎03(253)2464

ゲストOK!

Let's Music

MIDIプレイヤーAセット

CM-32L ¥69,000

SX-68M ¥19,800

Musicstudio Mu-1 ¥19,800

合計定価 ¥108,600

ツクモ特価 ¥91,800 (消費税別 ¥2,754)

クレジット例(税込)月々 ¥5,830 × 18回払

★Musicstudio PRO-68K V1.1又は、Music PRO 68K(MIDI)のソフトの場合には ¥8,000 プラスになります。

MIDIプレイヤーBセット

CM-64 ¥129,000

SX-68M ¥19,800

Musicstudio Mu-1 ¥19,800

合計定価 ¥168,600

ツクモ特価 ¥144,000 (消費税別 ¥4,320)

クレジット例(税込)月々 ¥7,107 × 24回払

ツクモグローバルカード

国内・外で活躍!!

使って便利、持って安心! ツクモグローバルカードはジャックス・VISA、セントラル・マスターとの提携カードです。ツクモ各店でお買物がらくらくできるうえに、国内はもとより海外でのショッピングもOK! しかも18才以上なら学生でもOK!

お申し込みは

(03)251-9898

又は各店で

18才以上なら
学生でもOK!



ツクモは「スーパーX PRO SHOP」です。

PRO STAFF

九十九電機株
〒101-91 東京都千代田区神田郵便局私書箱135号

★商品のご注文は在庫確認の上お願いします。



ツクモ7号店 ☎03-253-4199 (担当: 荒井)

便利で安心な通信販売

通信販売部 ☎03-251-9911

■ニューセンター店 ☎03-251-0987 (担当: 福地)

■ツクモ5号店 ☎03-251-0531 (担当: 川名)

■名古屋1号店 ☎052-263-1655 (担当: 吉高)

■名古屋2号店 ☎052-251-3399 (担当: 横山)

■ツクモ札幌 ☎011-241-2299 (担当: 村井)

カード払い

通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックスを御本人様より電話で通信販売部へお申し込み下さい。

全国代金引き換え配達

お申し込みは ☎03-251-9911へ

お電話1本!

配達日の指定もできます。

クレジット払い

月々 ¥3,000以上の均等払いも

頭金なし、夏・冬ボーナス2回

払いも受付中!

現金書留払い

〒101-91 東京都千代田区神田

郵便局私書箱135号

九十九電機株通信販売部 oh/X係

銀行振込払い

事前に ☎で届け先をご連絡下さい。

富士銀行 神田支店(普) №894047

九十九電機株

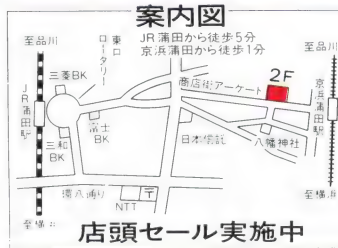
各種リース払い

くわしくは各店にお問い合わせ

下さい。ケースに合わせてご

相談にのらせて頂きます。

■平成2年夏のボーナス一括払い(7月末・8月末)OK!!手数料ナシ!!おトクです。ぜひ!!超低金利クレジットをご利用下さい。



90 オクトで始まるパソコンワールド

03-730-6271

●営業時間 AM 11:00 ~ 9:00/日曜・祭日PM7:00 電話一本で、ハイ即納
〒144 東京都大田区蒲田4-6-7 FAX 03-730-6273

全国通販

●定休日毎週火曜日 祭日の場合翌日になります。

オクト
ラクラククレジット

1回	2%	3回	2.5%	6回	3.5%	10回	5%	12回	5%	15回	7.5%
18回	9%	20回	10%	24回	11%	30回	14.5%	36回	15.5%	48回	20%

OCT-1 システム インフォメーション

- ▶全商品保証付(メーカー保証)
- ▶超低金利ハッピークレジット(1回~60回)頭金ナシOK!
- ▶ボーナス一括払いOK!ボーナス2回払いOK!!
- ▶配達日の指定OK!(万全なサポート体制)
- ▶商品の組合せ自由! オクトフリースタイルシステム
- ▶店頭デモンストレーション実施中

オクト
セレクトシステム

広告掲載商品以外の
製品も取扱っております。



浦田

●平成2年、夏のボーナス一括払い(手数料ナシ)
OKだよ〜ん。超低金利 ハッピークレジットです!!
EXPERT II・PROII 新発売記念セール開催中!!

OPEN

★下記セットでお買い上げの方にはプレゼント!! ●①MD-2HD 10枚②ジョイカード 2個(連射式)③シリコンキーボードカバー

お好みのセットをお選び下さい。 15型カラーディスプレイTV

- SX-WINDOW搭載。
- 40Mバイトハードディスク搭載

送料無料



EXPERT II・EXPERT II-HD

- CZ-603C-BK/GY 定価 ¥338,000
- CZ-613C-BK/GY 定価 ¥448,000

現金特価!! 推選
お電話下さい。

- SX-WINDOW搭載。
- 拡張I/Oポート4スロット装備



PROII・PROII-HD

- CZ-653C-BK/GY 定価 ¥285,000
- CZ-663C-BK/GY 定価 ¥395,000

CZ-8NJ2
●インテリジェントコントローラ
定価 ¥23,800
超特価 **¥18,800**



CZ-605D-GY/BK
定価 ¥115,000



CZ-613D-GY/BK
定価 ¥135,000

14型カラーディスプレイ



CZ-604D-GY/BK
定価 ¥94,800

21型カラーディスプレイ



CU-21HD
定価 ¥148,000

A CZ-603C + CZ-605D.....定価合計 ¥453,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

B CZ-613C + CZ-605D.....定価合計 ¥563,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

C CZ-653C + CZ-605D.....定価合計 ¥400,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

D CZ-663C + CZ-605D.....定価合計 ¥510,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

E CZ-603C + CZ-613D.....定価合計 ¥473,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

F CZ-613C + CZ-613D.....定価合計 ¥583,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

G CZ-653C + CZ-613D.....定価合計 ¥420,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

H CZ-663C + CZ-613D.....定価合計 ¥530,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

I CZ-603C + CZ-604D.....定価合計 ¥429,800 ▶ **オクト大特価**

12回	¥28,000	24回	¥14,800	36回	¥10,200	48回	¥8,000
-----	---------	-----	---------	-----	---------	-----	--------

J CZ-613C + CZ-604D.....定価合計 ¥542,000 ▶ **オクト大特価**

12回	¥36,000	24回	¥19,000	36回	¥13,100	48回	¥10,200
-----	---------	-----	---------	-----	---------	-----	---------

K CZ-653C + CZ-604D.....定価合計 ¥379,800 ▶ **オクト大特価**

12回	¥25,400	24回	¥13,400	36回	¥9,300	48回	¥7,200
-----	---------	-----	---------	-----	--------	-----	--------

L CZ-663C + CZ-604D.....定価合計 ¥489,800 ▶ **オクト大特価**

12回	¥32,200	24回	¥17,000	36回	¥11,800	48回	¥9,200
-----	---------	-----	---------	-----	---------	-----	--------

M CZ-603C + CU-21HD.....定価合計 ¥486,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

N CZ-613C + CU-21HD.....定価合計 ¥596,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

O CZ-653C + CU-21HD.....定価合計 ¥433,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

P CZ-663C + CU-21HD.....定価合計 ¥543,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

♡どんどんTELしよう。安くなるかもヨ!!

♡クレジット価格は、消費税込みですヨ。ご利用下さい!!

※クレジットの回数は1回~60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット:送料無料 ●店頭デモ実施中...専門の係員が詳細にアドバイス致します。ぜひご来店下さい。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

■店頭にて、ゲームソフト25%OFF! (税別) 超低金利 ハッピークレジットをご利用下さい。
■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の製品も取扱っております。

チャンス! X68000・SUPER-HD(チタン)=6月末発売/予約受付中!!

送料 ¥2,000

オクト限定スペシャルセット

SX-WINDOW搭載。



●ザ・ワークステーションと呼ぶにふさわしいスーパーな68000!! 新登場!!
SUPER-HD。

※プレゼント/①MD-2HD10枚 ③ジョイカード(連射式)
②アフターバーナー(¥9,200) ④シリコンキーボード(¥2,800)

X68000 SUPER-HD

●CZ-623C-TN+CZ-613D-TN

定価合計¥633,000... **大特価!! TEL下さい。**

※マウス・トラックボール付!! ディスプレイにはスピーカ2個、チルト台付!!

12回 ? 24回 ? 36回 ? 48回 ?

♡安くてゴメンなさい。今だけヨ!!

他のディスプレイ① CZ-602D、② 612D、③ CZ-603D、④ CU-21HDの組合せもごさいますのでお問い合わせ下さい。

※超低金利クレジットご利用下さい。1回~60回払い、頭金ナシ/ボーナス1回払い、ボーナス2回払いOK!

限定スペシャルセット (送料・消費税込み!!)
X68000EXPERT-HD



ラストチャンス!! 早い者勝ち!!

- CZ-612C (BK) (¥466,000)
- CZ-602D (BK) (¥99,800)
- MD-2HD 10枚
- ジョイカード(連射式×2個)
- ゲーム

オクト超特価
¥364,000

※ディスプレイ=①CZ-604D ②CZ-605D ③CZ-613D ④CU-21HD
との組合せもごさいます。TEL下さい。

オクト特選 シャープ周辺機器 (送料¥1,000)

- CZ-6BE1 IBM増設RAMボード (¥35,000) ▶ **特価 ¥26,500**
- CZ-6BE1B IMB増設RAMボード (¥28,000) ▶ **特価 ¥21,000**
- CZ-6BE2 2MB増設RAMボード (¥79,800) ▶ **特価 ¥60,500**
- CZ-6BE4 4MB増設RAMボード (¥138,000) ▶ **特価 ¥104,800**
- CZ-6BF1 増設用RS-232Cボード (¥49,800) ▶ **特価 ¥38,500**
- CZ-6BG1 GP-IBボード (¥59,800) ▶ **特価 ¥45,000**
- CZ-6BM1 MIDIボード (¥26,800) ▶ **特価 ¥20,500**
- CZ-6BN1 スキャナ用パラレルボード (¥29,800) ▶ **特価 ¥22,800**
- CZ-6BP1 数値演算プロセッサボード (¥79,800) ▶ **特価 ¥60,500**
- CZ-6B01 ユニバーサルI/Oボード (¥39,800) ▶ **特価 ¥30,500**
- CZ-6EB1/BK 拡張I/Oボックス (¥88,000) ▶ **特価 ¥66,800**
- CZ-6VT1/BK カラーイメージユニット (¥69,800) ▶ **特価 ¥53,000**
- CZ-6BL1 LANボード (¥268,000) ▶ **大特価**

- CZ-8NM2A マウス (¥68,800) ▶ **特価 ¥5,300**
- CZ-8NT1 マウストラックボール (¥98,800) ▶ **特価 ¥7,500**
- CZ-8NS1 カラーイメージスキャナ (¥188,000) ▶ **大特価**
- CZ-6BC1 FAXボード (¥79,800) ▶ **特価 ¥60,500**
- CZ-8TM2 モデムユニット (¥49,800) ▶ **特価 ¥38,000**
- CZ-64H 増設ハードディスク (¥120,000) ▶ **大特価**
- CZ-6TU GY/BK RGBシステムチューナー (¥33,100) ▶ **特価 ¥25,000**
- BF-68PRO 高性能CRTフィルター (¥19,800) ▶ **特価 ¥15,500**
- SX-68M(システムサコム) MIDIボード (¥19,800) ▶ **特価 ¥15,000**
- PI0-68BE1-A(I/O DATA) IMB増設RAMボード (¥25,000) ▶ **特価 ¥18,500**
- PI0-6BE2-2M(I/O DATA) 2MB増設RAMボード (¥50,000) ▶ **特価 ¥37,000**
- PI0-6BE4-4M(I/O DATA) 3MB増設RAMボード (¥88,000) ▶ **特価 ¥65,000**

オクト面白グッズ

アイテック(送料¥1,000)

- IT-X640 (¥158,000) ▶ **特価 ¥103,000**
- IT-X680 (¥198,000) ▶ **特価 ¥134,000**

モデムコーナー(送料¥1,000)

- MD-I200AIII... **特価 ¥14,800**
- MD-24FS4... **特価 ¥31,500**
- MD-24FS5... **特価 ¥34,800**
- MD-24FP4... **特価 ¥27,900**
- MD-I2FS... **特価 ¥15,000**

熱転写カラー漢字プリンター (ケーブル付) 送料¥1,000

CZ-8PC4 ¥99,800

限定

●48ドット

●サーマルヘッド

●B5~B4まで

●ハガキ可能

●カラー対応

オクト大特価 ¥56,800



- ① CZ-8PC3 (24ドット熱転写カラー漢字プリンター) 定価 ¥65,800... **特価 ¥45,000**
- ② CZ-8PK9 (24ピン漢字プリンター80桁) 定価 ¥89,800... **大特価!! TEL下さい。**
- ③ CZ-8PK10 (24ピン漢字プリンター136桁) 定価 ¥97,800... **大特価!! TEL下さい。**
- ④ CZ-8PG1 (24ピンカラー漢字プリンター80桁) 定価 ¥130,000... **大特価!! TEL下さい。**
- ⑤ CZ-8PG2 (24ピンカラー漢字プリンター136桁) 定価 ¥160,000... **大特価!! TEL下さい。**
- ⑥ IO-735X (カラーイメージジェット) 定価 ¥248,000... **大特価!! TEL下さい。**

パソコンラック 推奨 送料 無料

①五段キャスター付

②四段キャスター付



5段キャスター付
キーボードが収納できるから、手でマウス操作がラクできる
棚板5段のマルチに活用できるデスク。
ワゴン、こいつはデキル!
1325(H)×640(W)×700(D)
特価 ¥16,000



4段キャスター付
どんなパソコンにもフレキシブルに対応! 使い易いデスクです。
1245(H)×614(W)×600(D)
特価 ¥12,000

X68000ソフト大セール実施中※ゲームソフトオール25%off

<グラフィック> ●Z's STAFF PRO68K Ver.2.0 (シャフト) 定価 ¥58,000

オクト特価 ¥40,000

<データベース> ●KAMIKAZE (サムシンググッド) 定価 ¥68,000

オクト特価 ¥46,000

<グラフィック> ●C-TRACE68 (キャスト) 定価 ¥68,000

オクト特価 ¥51,000

<言語> ●C & Professional Pack (マイクロウェアジャパン) 定価 ¥58,000

オクト特価 ¥44,000

<グラフィック> ●サイクロン エキスプレス 定価 ¥78,000

オクト特価 ¥58,000

<グラフィック> ●デジタルクラブ 定価 ¥39,800

オクト特価 ¥28,000

<ワープロ> ●ハイパーワード 定価 ¥39,800 CZ-251BS

オクト特価 ¥29,800

型 名	商 品	定 価	特 価
CZ-211LS	Ccompiler PRO-68K	¥39,800	¥28,800
CZ-212BS	BUSINESS PRO-68K	¥68,000	¥48,000
CZ-213MS	MUSIC PRO-68K	¥18,800	¥13,500
CZ-214MS	SOUND PRO-68K	¥15,800	¥11,500
CZ-215MS	Sampling PRO-68K	¥17,800	¥12,800
CZ-219SS	OS-9/X68000	¥29,800	¥21,000
CZ-220BS	DATA PRO-68K	¥58,000	¥41,000
CZ-221HS	New Print Shop PRO-68K	¥19,800	¥14,300
CZ-223CS	Communication PRO-68K	¥19,800	¥14,300
CZ-224LS	THE 福袋 V2.0	¥9,900	¥7,500
CZ-226BS	CARD PRO-68K	¥29,800	¥21,300
CZ-241BS	システム手帳リフィル集	¥9,800	¥7,500
CZ-242BS	活用フォーム集	¥9,800	¥7,500
CZ-244SS	Homan 68K Ver.2.0	¥9,800	¥7,500
CZ-247MS	MUSIC PRO-68K (MIDI)	¥28,800	¥20,800
CZ-240BS	Stationery PRO-68K	¥14,800	¥11,500
CZ-243BS	CYBER NOTE PRO-68K	¥19,800	¥15,200
EW		¥38,000	¥29,800
G-68K		¥14,800	¥11,400
E-68K		¥19,800	¥15,300

店頭ゲームソフトオール25%off! ビジネスソフト 25%より特価中

●尚、送料として1ヶ ¥500、2ヶ ¥700、3ヶ以上で ¥1,000 となります。(税別)

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL: 03-730-6271

お申込みは電話でお願いします。お客様の住所・氏名・電話番号及び商品名をお知らせ下さい。●入金確認後、商品をご送付いたします。

現金振込

銀行振込: お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留: 封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット

専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

1回	2%	3回	2.5%	6回	3.5%	10回	5%
12回	5%	15回	7.5%	18回	9%	20回	10%
24回	11%	30回	14.5%	36回	15.5%	48回	20%

振込先

富士銀行 三菱銀行
久ヶ原支店 蒲田支店
④No.1824 ④No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。

※連休のお知らせ=7/31(水)、8/1(水)は連休です。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。

※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

平成2年夏のボーナス一括払いOK!! (7月末・8月末) 手数料ナシ!! 超低金利クレジットをご利用下さい。

注目!!

夏のボーナス一括払い
手数料(金利)無料
(7月末・8月末どちらかご指定下さい)

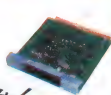
モデム(AIWA) 50台限定 (送料¥1,000)
PV-A24MNP5 (定価 ¥54,800)
●MNPクラス5 限定特価¥26,500
●2400bps (送料・消費税込 ¥28,325)

CYBER STICK

●CZ-8NJ2
(定価 ¥23,800)
超特価!!
▶価格はTEL下さい



X68000シリーズ専用 特価 ¥16,480
MIDIインターフェースボード
SX-68M (サコム)
(純生コンパチ) 定価 ¥19,800
送料・消費税込み!!



6/15~7/15

X-1ターボZIII 特別ご提供品!!

台数限定

●CZ-888C+CZ-860D+M-2HD(10枚)
定価 ¥269,600 ▶特価 ¥164,800
(ボーナス併用も有りますTEL下さい)

・ジョイカード
・ゲーム3種
・パソコンラック(A)3段
プレゼント中
送料消費税込み!!

12回	14,400	24回	7,600	36回	5,300	48回	4,100	60回	3,400
-----	--------	-----	-------	-----	-------	-----	-------	-----	-------

- お近くの方はお
- 本体単品で特
- ビジネスソフト定

ジョイスティック 送料 ¥500
●X-1PRO
定価 ¥9,500 ▶特価 ¥7,800
●ASCII STICK
定価 ¥6,800 ▶特価 ¥5,500

NEW X68000EXPERT II/II-HD & PROII/PROII-HD & SUPER-HD (送料・消費税込)

EXPERT II

セットでお買い上げの方に、
●ディスク10枚 } プレゼント中!!
●ゲーム3種
●ジョイカード2枚

Aセット: CZ-603C+CZ-604D	定価 ¥432,800 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
Bセット: CZ-603C+CZ-605D	定価 ¥453,000 ▶特価 (価格はお電話下さい)
12回 30,200 24回 15,900 36回 11,000 48回 8,500 60回 7,100	
Cセット: CZ-603C+CZ-613D	定価 ¥473,000 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
Dセット: CZ-603C+CU-21HD	定価 ¥486,000 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

EXPERT II-HD

セットでお買い上げの方に、
●ディスク10枚 } プレゼント中!!
●ゲーム3種
●ジョイカード2枚

Aセット: CZ-613C+CZ-604D	定価 ¥542,800 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
Bセット: CZ-613C+CZ-605D	定価 ¥563,000 ▶特価 (価格はお電話下さい)
12回 37,700 24回 19,800 36回 13,700 48回 10,600 60回 8,900	
Cセット: CZ-613C+CZ-613D	定価 ¥583,000 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
Dセット: CZ-613C+CU-21HD	定価 ¥596,000 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

PRO II

セットでお買い上げの方に、
●ディスク10枚 } プレゼント中!!
●ゲーム3種
●ジョイカード2枚

Aセット: CZ-653C+CZ-604D	定価 ¥379,800 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
Bセット: CZ-653C+CZ-605D	定価 ¥400,000 ▶特価 (価格はお電話下さい)
12回 26,800 24回 14,100 36回 9,700 48回 7,600 60回 6,300	
Cセット: CZ-653C+CZ-613D	定価 ¥420,000 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
Dセット: CZ-653C+CU-21HD	定価 ¥433,000 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

PRO II-HD

セットでお買い上げの方に、
●ディスク10枚 } プレゼント中!!
●ゲーム3種
●ジョイカード2枚

Aセット: CZ-663C+CZ-604D	定価 ¥489,800 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
Bセット: CZ-663C+CZ-605D	定価 ¥510,000 ▶特価 (価格はお電話下さい)
12回 34,100 24回 17,900 36回 12,400 48回 9,500 60回 8,100	
Cセット: CZ-663C+CZ-613D	定価 ¥530,000 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
Dセット: CZ-663C+CU-21HD	定価 ¥543,000 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

SUPER-HD

セットでお買い上げの方に、
●ディスク10枚 } プレゼント中!!
●ゲーム3種
●ジョイカード2枚

Aセット: CZ-623TN+CZ-604D	定価 ¥592,800 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
Bセット: CZ-623TN+CZ-605D	定価 ¥613,000 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	
Cセット: CZ-623TN+CZ-613D	定価 ¥633,000 ▶特価 (価格はお電話下さい)
12回 42,700 24回 22,500 36回 15,500 48回 12,100 60回 10,100	
Dセット: CZ-623TN+CU-21HD	定価 ¥646,000 ▶特価 (価格はお電話下さい)
12回 ? 24回 ? 36回 ? 48回 ? 60回 ?	

X68000シリーズ ~P&Aスペシャルセット=限定誌上販売!!

台数限定 送料、消費税込み

セットでお買い上げの方に、
●ディスク10枚 ●ゲーム3種 ●ジョイカード2個 プレゼント中

EXPERT

●CZ-602C+CZ-612D 定価 ¥475,800 ▶特価 ¥306,000
●CZ-602C+CZ-604D 定価 ¥450,800 ▶特価 ¥300,000
●CZ-602C+CZ-605D 定価 ¥471,000 ▶特価 ¥320,000
●CZ-602C+CZ-613D 定価 ¥491,000 ▶特価 ¥336,000
●CZ-602C+CU-21HD 定価 ¥504,000 ▶特価 ¥338,000

EXPERT-HD

●CZ-612C+CZ-612D 定価 ¥585,800 ▶特価 ¥375,000
●CZ-612C+CZ-604D 定価 ¥560,800 ▶特価 ¥369,000
●CZ-612C+CZ-605D 定価 ¥581,000 ▶特価 ¥386,000
●CZ-612C+CZ-613D 定価 ¥601,000 ▶特価 ¥403,000
●CZ-612C+CU-21HD 定価 ¥614,000 ▶特価 ¥407,000

PRO-HD

●CZ-662C+CZ-612D 定価 ¥527,800 ▶特価 ¥339,000
●CZ-662C+CZ-604D 定価 ¥502,800 ▶特価 ¥333,000
●CZ-662C+CZ-605D 定価 ¥523,000 ▶特価 ¥352,000
●CZ-662C+CZ-613D 定価 ¥543,000 ▶特価 ¥368,000
●CZ-662C+CU-21HD 定価 ¥556,000 ▶特価 ¥372,000

回〜84回払いまでOK!!

★頭金なし!★即日発送

がスバリ超特価セールでご奉仕!!

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

立寄り下さい。専門係員が説明いたします。
価で受付します。詳しくは電話にてお問合せ下さい。
価の20%引きOK! TELください。

全国通販

X68000用ソフトコーナー(送料1ヶ〜5ヶまで¥500)

Z's STAFF PRO68K Ver.2.0(ツァイト)	定価 ¥ 58,000	特価 ¥ 39,700
Z's TRIPHONY デジタルクラフト(ツァイト)	定価 ¥ 39,800	特価 ¥ 29,300
テラツツ(ハンギングバード)	定価 ¥ 19,800	特価 ¥ 15,800
KAMIKAZE (サムシング・グッド)	定価 ¥ 68,800	特価 ¥ 46,000
EW & EI (イースト)	定価 ¥ 38,800	特価 ¥ 28,800
C & Professional Pack (マイクローブアジアペン)	定価 ¥ 58,800	特価 ¥ 43,000
Final Ver.3.2 (エーエスピー)	定価 ¥ 38,000	特価 ¥ 30,000
DATA PRO68K CZ220BS	定価 ¥ 58,000	特価 ¥ P&A 特価
CARD PRO68K CZ220BS	定価 ¥ 29,800	TEL下さい!
C-computer PRO68K CZ21ILS	定価 ¥ 39,800	特価 ¥ 32,000
OS-9/X68000 CZ219SS	定価 ¥ 29,800	特価 ¥ P&A 特価 TEL下さい!
AI-68K CZ234LS	定価 ¥ 188,000	特価 ¥ 143,000
THE 編集 V2.0 CZ224LS	定価 ¥ 9,900	特価 ¥ 7,700
SOUND PRO68K	定価 ¥ 15,800	特価 ¥ 12,500
MUSIC PRO68K CZ213MS	定価 ¥ 15,800	特価 ¥ P&A 特価 TEL下さい!
Sampling PRO68K CZ215MS	定価 ¥ 17,800	特価 ¥ 14,000
MUSIC-studio PRO68K 237MS	定価 ¥ 15,800	特価 ¥ P&A 特価 TEL下さい!
MUSIC-PRO68K (MIDI) 247MS	定価 ¥ 28,800	特価 ¥ 22,000
New-grant Ship 221HS	定価 ¥ 19,800	特価 ¥ P&A 特価
Communication 223CS	定価 ¥ 19,800	特価 ¥ TEL下さい!
C-TRACE68 Ver.3.0 (キャスト)	定価 ¥ 98,000	特価 ¥ 77,000
サクロン EXPRESS α68	定価 ¥ 98,000	特価 ¥ 72,000
068K Ver.2 PRO	定価 ¥ 22,000	特価 ¥ 16,300
THE FILE PROFESSOR (ロゴシステム)	定価 ¥ 28,000	特価 ¥ 20,500
ツール (サンソフト)	定価 ¥ 28,000	特価 ¥ 20,500
た〜みのる2 (SPS)	定価 ¥ 17,800	特価 ¥ 13,500
マジックハット (ミュージカルプラン)	定価 ¥ 19,800	特価 ¥ 14,900
Hyper word CZ21BS	定価 ¥ 39,800	特価 ¥ 30,900

●ゲームソフト20%OFF OK!! (一部ソフト除く)

周辺機器コーナー(送料 ¥1,000)

A CZ-8NSI	定価 ¥ 188,000	特価 ¥ 145,000
B CZ-6VTI	定価 ¥ 69,800	特価 ¥ 54,000
C CZ-6TU	定価 ¥ 33,100	特価 ¥ 25,000
D BF-68PRO	定価 ¥ 19,800	特価 ¥ 15,500
E CZ-6BEI	定価 ¥ 35,000	特価 ¥ 26,500
F CZ-6BEIA	定価 ¥ 38,000	特価 ¥ 28,600
G CZ-6BE2	定価 ¥ 79,800	特価 ¥ 60,000
H CZ-6BE4	定価 ¥ 138,000	特価 ¥ 107,000
I CZ-6BFI	定価 ¥ 49,800	特価 ¥ 38,200
J CZ-6BPI	定価 ¥ 79,800	特価 ¥ 61,000
K CZ-6BMI	定価 ¥ 26,800	特価 ¥ 20,300
L CZ-6EBI	定価 ¥ 88,000	特価 ¥ 67,500
MAN-S100	定価 ¥ 36,600	特価 ¥ 28,500
N CZ-6SDI	定価 ¥ 44,800	特価 ¥ 35,000
O CZ-8PC3	定価 ¥ 65,800	
P CZ-8PC4	定価 ¥ 99,800	
Q CZ-8PG1	定価 ¥ 130,000	
R CZ-8PG2	定価 ¥ 160,000	
S CZ-8PK10	定価 ¥ 97,800	
T CZ-6PVI	定価 ¥ 198,000	特価 ¥ 153,000
UIO-735X	定価 ¥ 248,000	特価 ¥ 190,000
V CZ-8BSI	定価 ¥ 23,800	特価 ¥ 19,000
W PIO-6BEI-A(I/O DATA)	定価 ¥ 25,000	特価 ¥ 18,200
X PIO-6BE2-2M(I/O DATA)	定価 ¥ 50,000	特価 ¥ 36,800
Y PIO-6BE4-4M(I/O DATA)	定価 ¥ 88,000	特価 ¥ 64,800

P&A超特価
TEL下さい。

中古パソコンはP&Aにおまかせ!!

その場で高価現金買取・高価下取りOK!!

- まずはお電話下さい。 03-651-1884 FAX:03-651-0141
- 下取りの場合 価格は常に変動していますので査定額をお電話で確認して下さい。(差額は、P&A超低金利クレジットをご利用下さい。)
- 買取の場合 現品が着き次第、2日以内に買取金額を連絡し、振込み、又は書留でお送り致します。
- 近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、¥1,000,000までお支払い致します。

《便利な超低金利クレジットをご利用下さい》

- 月々¥1,000円からOK!! ●ボーナス払いOK(夏冬10回までOK)
- 支払い回数 1回〜84回 ●お支払いは、8ヶ月先からでもOK!!

アフターサービス完全

全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブル等。
万が一初期不良、輸送トラブルが発生した際には、即交換させていただきます。

●定休日/毎週水曜日=第3水曜・木曜は連休とさせていただきます(祭日の場合は翌日になります)

X68000用ハードディスク(送料 ¥1,000)

アイテム

- HXD-040(40MB/23ms) 定価 ¥118,000 特価 ¥ 88,000
- HXD-042(増設用) 定価 ¥128,000 特価 ¥ 95,000

アイテム

- ITX-640(40MB/28ms) 定価 ¥158,000 特価 ¥101,000
- ITX-680(80MB/20ms) 定価 ¥198,000 特価 ¥131,000

プリンター(ケーブル・用紙付)限定5台 新品(送料 ¥1,000)

- CZ-8PC3(カラー漢字24ドット熱転写プリンター)
定価 ¥65,800 特価 ¥39,800
- CZ-8PK8(24ピン漢字プリンター136桁)
定価 ¥152,000 特価 ¥75,800
- CZ-8PC4 P&A特選!!(カラー漢字48ドット熱転写プリンター)
定価 ¥99,800 特価 ¥59,000

モデムコーナー(送料 ¥1,000)

- (A) MD-24FS5(オムロン) 定価 ¥ 49,800 特価 ¥ 34,800
- (B) MD-24FS7(オムロン) 定価 ¥ 64,800 特価 ¥ 45,000
- (C) コムスター2424/4(NEC) 定価 ¥ 38,800 特価 ¥ 28,000
- (D) コムスター2424/5(NEC) 定価 ¥ 44,800 特価 ¥ 32,000

P & A 特選パソコンラック(送料無料で移動自由(キャスター付))

③3段	④4段	⑤5段
875(H) ×580(D) ×610(W)	1320(H) ×600(D) ×630(W)	1280(H) ×600(D) ×620(W)
¥9,000	¥12,000	¥15,000

中古パソコン 送料 ¥2,000

- X-68000セット 定価 ¥210,000 特価 ¥150,000
- X-68000ACEセット 定価 ¥240,000 特価 ¥180,000
- X-1ターボセット 定価 ¥100,000 特価 ¥75,000
- X-1G/30セット 定価 ¥39,000 特価 ¥29,000
- CZ-822C 定価 ¥15,000 特価 ¥10,000
- CZ-830C 定価 ¥25,000 特価 ¥19,000
- CZ-856C 定価 ¥45,000 特価 ¥34,000
- CZ-870C 定価 ¥55,000 特価 ¥42,000
- CZ-881C 定価 ¥65,000 特価 ¥50,000
- CZ-820D 定価 ¥10,000 特価 ¥7,000
- CU-14AG2 定価 ¥30,000 特価 ¥22,000
- CU-14H2 定価 ¥30,000 特価 ¥22,000
- CZ-8PC2 定価 ¥25,000 特価 ¥19,000
- CZ-8PK6 定価 ¥32,000 特価 ¥24,000
- CU-14GB 定価 ¥5,000 特価 ¥3,000
- CU-14BD 定価 ¥25,000 特価 ¥19,000

通信販売お申し込みのご案内

[現金一括でお申し込みの方]

- 商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)

[銀行振込でお申し込みの方]

- 銀行振込ご希望の方は必ずお振込みの前に電話にてお客様の住所・お名前・商品名等をお知らせください。

(電話扱いでお振込み下さい。)

[振込先] 住友銀行 新小岩支店
当No.263914 株ビー・アンド・イー

[クレジットでお申し込みの方]

- 電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
- 現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
- 1回〜84回払いまで出来ます。但し、1回のお支払いは¥1000円以上。

超低金利クレジット率

回数	3	6	10	12	18	24	36	48	60	72	84
手数料	2.5	3.5	5.0	5.0	9.0	10.5	14.5	19.0	24.5	32.0	38.5



- マイコン
- ビデオ
- ビデオテープ

P&A

株式会社ピー・アンド・イー
〒124 東京都葛飾区新小岩2丁目1番地19号

☎03-651-0148(代) 03-651-0141

営業時間
平日:AM10:00~PM7:00
日祭:AM10:00~PM6:00

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合わせ下さい。

超特価でクレジットが組める!!

スケールが違う!

THE 日本一のコンピュータ館です COMPUTER館

より専門化されたビジネスシステムたち



●4F ビジネスシステムフロアは、様々な業種や業務別に専門化されたシステムがラインナップされています。例えば、卸・小売業に役立つ、店舗POSシステムや、製造業で使われる、部品や在庫の管理システム。さらには、教育関連の成績処理や翻訳、法律情報システムと言ったぐあいにこの他にもあらゆる業種に対応するシステムが体験できます。また業務別では、会社組織の様々な部署で役立つ、ソフトを含めたシステムが体験できるようになっています。



パソコンソフト・パソコン書籍30,000点展示



●建築・機械設計のお仕事に携わっている方にお奨めしたいのが、正確で美しい図面を素早く作製できるCADシステムです。さらに、急な図面の変更にも対応し、高性能プロッタで美しく作製なおせます。また、このフロアには、コンピュータグラフィックスやCAMと言って、小さな彫刻物のサンプルを製作する優れたシステムもございます。

ザンピュータ館フロアごあんない

6F

イベントフロア

催事場

5F

CAD&グラフィックスフロア

CAD・CAM・グラフィックスのソフトから周辺機器及びシステムまで「ワークステーション/モデム/NET/パソコン通信用機器及び体験コーナー」

4F

ビジネスシステムフロア

業種別パソコンコーナー（小売業、不動産業、製造業、教育、広告、飲食店など業種にあわせたソフト及びシステムを動かし実際に体験できるコーナーです）業務別パソコンコーナー（社長室、営業部、総務部、経理部、OA推進部など業務にあわせたソフト及びシステムを動かし実際に体験できるコーナーです）ラップトップパソコン 周辺機器 機密 セミナールーム

3F

ビジネスソフトフロア

ビジネスソフト/帳票
IBM/J-3100/AX/Macintosh

2F

アクセサリ&ワープロフロア

ワープロ/ワープロ周辺機器/電子手帳/OAシステムインテリ/アコビ/ファクシミリ/アクセサリ/サブライ/セミナールーム

1F

ホビー&BOOKフロア

パソコン関連書籍/雑誌/ホビーパソコン/周辺機器/ゲーム/ホビーソフト/プレイゾーン

●秋葉原電気まつり「THE AKIHABARA EXCITING FESTA」開催。期間/6月21日木～7月22日日 ※抽選で「ジャパニーズ・ロマンチック・クルーズ」をペアでご招待。

情報鮮度が違う!

日本のガリバー「PC-9801」展

期間/6月15日(金)～7月1日(日)・会場/6F イベントフロア

①PC-9801オールマシン展

現在活躍中のPC-9801シリーズのフルラインナップを始めとして、モニター・ハードディスク・プリンタなどの周辺機器を集中展示いたします。98本体はもちろんのこと、周辺機器選びでお困りの方にはうってつけの催しです。

②PC-9801ペリフェラル展(周辺機器サードパーティー大集合)

PC-9801シリーズと、その関連ソフトを100%活用するための周辺機器が大集合。プリンタ、ハードディスク、各種ボード、キーボード、マウスなど、あなたのシステム環境をさらによりよくするために役立ててください。

③PC-9801 スーパーセミナー

■期間/6月15日(金)～21日(木)

①MS-DOS入門セミナー(6/15(金)～17日(日)) ①13:30～14:30 ②15:00～16:00 ③16:30～17:30

●MS-DOSとは何か?から始まり、MS-DOSの歴史まで、MS-DOSの基礎知識を身につけていただくためのセミナーです。

②PC-9801入門セミナー(6/18(月)・19(火)) ①13:30～14:30 ②15:00～16:00 ③16:30～17:30

●何げなく使われている、bit・RAM・ROM・クロックなどの専門用語を始めとした、98をより納得してご利用していただくためのセミナーです。

③周辺機器セミナー(6/20(水)・21(木))

20日① ①13:30～14:30(ハードディスク) ②15:00～16:00(EMSボード) ③16:30～17:30(ハードディスク)

21日① ①13:30～14:30(EMSボード) ②15:00～16:00(ハードディスク) ③16:30～17:30(EMSボード)

●PC-9801シリーズに使われる、ハードディスクとEMSボードの基礎知識から、応用に至るまでをわかり易く解説するセミナーです。

※スーパーセミナーのお申し込み、お問い合わせは、当店係員にお申し付けください。多数のお申し込みが予想されますのでお早め。

●豊富なソフトウェアに、豊富な周辺機器で数多くのユーザーに愛されるPC-9801シリーズ。その数々の98の様々な面に出会える楽しいつばいの17日間です。98ユーザーの方も、これから購入の方も、目が離せない「PC-9801」展です。

④98アプリケーション活用クリニック

■期間/6月22日(金)～24日(日)

PC-9801シリーズ対応の著名ソフトをうまく活用しきれない、又は活用方法がわからないなどの理由で、お困りの方/直接ご自身でソフトハウスに、自分の使用方法についてのアドバイスが得られるチャンスです。尚、ソフトハウスは、限られておりますので、詳細につきましては当店係員にお尋ね下さい。

⑤ゲーム大会

■期間/6月29日(金)～7月1日(日)

PC-9801シリーズ対応の、今、話題のゲームソフトで、エキサイティングな大会を行います。奮ってご参加ください。

※日本のガリバー「PC-9801」展の詳細につきましては、当店6Fイベント係りに、お尋ね下さい。

パソコンソフトセミナーのご案内

月 日	時 間	セミナー名	ソフトハウス	ジャンル	会場
6月18日(月)	17:00～19:00	商 魂	PCA	販売管理	4F
6月19日(火)	17:00～19:00	アシストカード(PC9801用)	アシスト	データベース	4F
6月20日(水)	17:00～19:00	KOAテクノメイト	高電社	ワープロ	4F
6月21日(木)	17:00～19:00	Dyna CAD+Dyna PERS3	ダイナウェア	CAD	4F
6月24日(日)	14:00～16:00	ロゴライター	DTP	ロ ゴ	4F
6月25日(月)	17:00～19:00	初めてさわるTOP財務会計	オービック	会 計	4F
6月26日(火)	17:00～19:00	UP2システム	ダイナウェア	総合型	4F
6月27日(水)	17:00～19:00	MS-WINDOWS体験セミナー	日本ソフトバンク	ウインドウズ	4F
6月28日(木)	17:00～19:00	Aldus Page Maker	日本ソフトバンク	DTP	4F
6月29日(金)	17:00～19:00	マイクロソフトエクセル	日本ソフトバンク	表計算	4F
7月1日(日)	14:00～16:00	毛筆わーぶろ(入門)	富士ソフトウェア	ワープロ	4F
7月2日(月)	17:00～19:00	ロータス1・2・3紹介コース	ロータス	表計算	4F
7月3日(火)	17:00～19:00	PI.EXE ワープロ入門	dBソフト	ワープロ	4F
7月5日(木)	17:00～19:00	初めてさわる一太郎	ジャストシステム	ワープロ	4F
7月6日(金)	17:00～19:00	PCA会計II	PCA	会 計	4F
7月7日(土)	14:00～16:00	データベース入門 THE CARD3	アスキー	データベース	4F
7月8日(日)	14:00～16:00	EXCEL(初級編)	マイクロソフト	表計算	4F
7月9日(月)	13:00～15:30	ハードディスク入門	アイテック	ハード	4F
7月9日(月)	16:00～18:30	ハードディスク入門	アイテック	ハード	4F
7月10日(火)	15:00～17:00	How to社内報UPシリーズ	ダイナウェア	DTP	4F

※都合により、変更する場合がありますがその際はご了承ください。

ワープロセミナーのご案内

月 日	時 間	セミナー名	機 種	メーカー	会場
6月19日(火)	17:00～19:00	電子手帳セミナー	DK5000	カシオ	2F
6月21日(木)	14:00～16:00	RUPOセミナー中級	JW95G	東 芝	2F
6月25日(月)	16:00～18:00	キャノワードセミナー	ALPHA50	キャノン	2F
6月28日(木)	17:00～19:00	OASYSセミナー中級	OA30AXS	富士通	2F
7月4日(水)	17:00～19:00	シャープ新製品セミナー	WDA340	シャープ	2F
7月5日(木)	17:00～19:00	OASYSセミナー初級	OA30AXS	富士通	2F

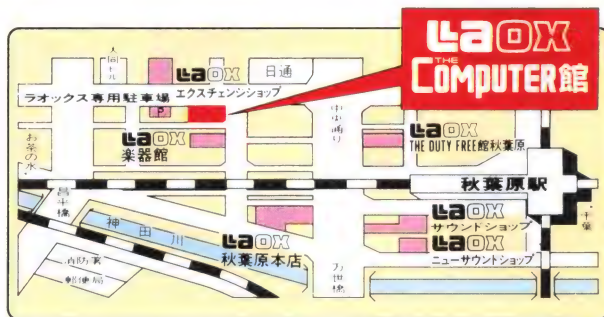
※都合により、変更する場合がございますがその際はご了承ください。

受講ご希望の方は

●各セミナーとも専門の係員が、実際に皆さんと共にパソコンやワープロを起動させながら説明していきます。ぜひ、この機会をご利用ください。尚、各セミナーとも定員になり次第締め切らせていただきますのでご了承ください。お申し込み、お問い合わせは、お電話にて当店セミナー係員にお申し付けください。多数のお申し込みが予想されますので、セミナー受講のご予約はお早めに願います。

その他、楽しく役立つイベントやソフトのデモ実演など最先端の情報がいっぱいです。

さらに、3,500名様にJRギフトカード1万円分をプレゼント!詳しくは当店係員まで。



ザ・コンピュータ館

TEL/03-5256-3111

〒101 東京都千代田区外神田1-7-6(秋葉原駅徒歩4分) P 駐車場完備

LaOX

1990年9月12日まで

おかげさまで5周年 アートディンク サンクスフェア

実施中

A賞

次のうちいずれかひとつを、
抽選で合計10名様にプレゼント。

NEC	98NOTE	3名様
緑電子	40MBハードディスク	3名様
シャープ	20MBハードディスク	1名様
エプソン	熱転写プリンタ	2名様
シャープ	熱転写プリンタ	1名様

B賞

オリジナルハードケース入フロッピーディスク
(5枚セット)を抽選で毎月100名様、
合計500名様にプレゼント。

応募方法

パッケージに同封されたユーザ登録ハガキに
必要事項をご記入のうえ当社宛ご返送
ください。

応募締切

1990年9月12日まで(当日消印有効)

発表・発送

A賞の当選発表は、当選者本人への通知、
及び10月発売のパソコン専門誌(「Oh!PC」
「Login」等)の誌上広告にて。
賞品発送は10月中旬です。B賞の当選発表
は、賞品の発送(毎月10日締切15日発送)
をもって、かえさせていただきます。限定
お買得版鉄道王の夢を追うオムニバスタイプ・シミュレーションゲーム。
マップ数は、なんと10!ほま世界中を手中にできる。

A列車で行こうII

●X68000 ●5"2HD(4枚組)

特別価格: 12,800円

30年のプロゴルファー人生を歩むロールプレイング付。
他に類をみない2WAYゴルフシミュレーション。2WAYゴルフシミュレーション
Double Eagle
ダブルイーグル

●X68000 ●5"2HD(2枚組)

標準価格: 9,500円

限定
お買得版太平洋戦争の激闘をモチーフにしたウォーシミュレーション。
6つのシナリオを収め、FD8枚組の大作バージョンで登場。

SUPER REAL TIME WAR SIMULATION GAME

大海令

大日本帝国海軍の軌跡

●X68000 ●5"2HD(8枚組)

特別価格: 13,800円

光の刃、陽子砲が星海に躍る。そして敵機を切り裂く。
太陽系の運命を賭けて戦うスペースウォーシミュレーション。FAR
SIDE MOON
You war in the last defense force

●X68000 ●5"2HD(3枚組)

標準価格: 9,500円

限定
お買得版ソロモン諸島近海を舞台にリアルタイムの局地戦を展開。
このバージョンは、合計12本のシナリオを収録した豪華版。

SUPER REAL TIME WAR SIMULATION GAME

南海の死闘

●X68000 ●5"2HD(4枚組)

特別価格: 9,500円

戦車72両、2,000を越す将兵を率いて、
無敵のドイツ機甲師団に挑む。TANK BATTLE SIMULATION
機甲師団

●今秋発売予定

表示価格に消費税は含まれません。

△68000ユーザの皆様へ

すでに、各報道でご承知のことと存じますが、4月13日に発売した初期出荷バージョン「ファーサイドムーンX68000版」(機種表示部分が真赤なもの)の一部にコンピュータウイルスが発見され、X68000ユーザの皆様のみならず、社会全体をおさわがせしましたことに深くお詫び申し上げます。

問題となっているウイルスは、X68000本体のS-RAMに付着する「NX68K IPL V1.02」という種類のもので、症状は今年の7月になったらドライブに差し込まれたディスクのデータを破壊するかも知れないという極めて悪質なものです。

しかし、このウイルスを撲滅することができる「ワクチン」がすでに準備されており、ただいま、弊社登録ユーザの皆様無料で配布いたしております。まだユーザ登録をされていないお客様は、大至急、当社まで「ユーザ登録ハガキ」をご返送ください。

未来あるパソコンライフ、安心してコンピュータに親むことができる社会環境のためにも、ユーザの皆様方の絶大なるご支援をお願いする次第です。何卒、ご理解ご協力をたまわりますよう、よろしくお願い申し上げます。

マイコンショウ'90& 第70回ビジネスショウ

例年どおり東京平和島にある流通センターでマイコンショウ'90、晴海国際見本市会場で第70回ビジネスショウが開催された。いずれも今後の世の中の動き、特にコンピュータ業界の趨勢を占う重要な催しといえるだ

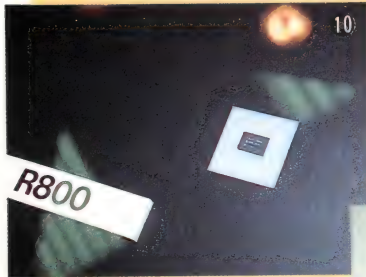
ろう。
今年から来年にかけての業界や新製品の動きを先取りする両ショウの内容をパソコン関係を中心にレポートしてみたい。

マイコンショウ



- ①ずらりと並んだカラー液晶
- ②ファミコン程度の絵なら小型液晶でも十分鮮明。反応も速い
- ③鮮明なTFTカラー液晶ディスプレイ
- ④これがマルチワード

- ⑤DATメモリ
- ⑥ICEを使ったゲーム開発
- ⑦ついに実物が見られた68040
- ⑧V70とAFPCのボード



- ⑨ 最近のオシロはとにかく凄い (ソニーテクノニクス)
- ⑩ これがR800
- ⑪ 電波新聞社。開発中のギャラガ'88が見える
- ⑫ TRONは元気だ
- ⑬ TX1を使ったラップトップB-TRON
- ⑭ なぜかJALもTRON。ウィンドウにはビデオが映る

ビジネスショウ



マイコンショウ

年々地味な様相を呈するマイコンショウ。大手メーカーではパソコン、液晶ディスプレイやカスタムLSI、ワンチップマイコン、各種コントローラなど、その他サードパーティによる開発システムや周辺機器が主な出展物だ。

シャープでは業界最強のカラー液晶を始め、大小充実した液晶ディスプレイ、エレクトロルミネッセンスディスプレイなどを展示。もはやお馴染みとなった14インチ液晶カラーテレビも、いまだ他社の追随を許していない。しかし、他社の追い上げも激しく、今回の出展では日立の10インチカラー液晶ディスプレイユニットが注目株だ。

X68000関係はX68000 SUPER-HD、SX

-WINDOWが中心。昨年登場していた W ORD PRO-68KはMultiWord と名前を変え、内容を一新して出展。高速テキスト画面と印刷イメージに忠実なウィンドウ画面を高速に切り換えて使用する。

CPU関係では、なんといってもR800。

R800はアスキーとヤマハの共同開発によるZ80と上位互換性があるという16ビットCPU。同クロックでもZ80の3倍以上の速度を持ち、乗算命令を追加、アドレス空間は16Mバイトという、Z80ファンなら思わずスキップしたくなる仕様。ちなみに最高29MHzで動作する。うーむ、MSX3でも出ないかな。

こういったショウでは初めてモトローラの68040が展示されていたり、NECの国産32ビットCPU、V70/80シリーズ用のAFP

C(高性能浮動小数点コプロセッサ)がようやくお目見えしたりと、見る人が見れば期待のプロセッサ群が発表されていた。

そのほか、最近のFAは凄いと思わせられたのがソニーテクノニクスのオシロスコープ。カラー画面を採用し、複雑な操作も画面のメニューに触るだけ。CRTに張られたタッチセンサと指の動きを捉える光センサのマトリクスが決め手だ。ペンツ1台分の値段というのが残念だが、家電製品にもこんな操作性があればと思わせる。

ビジネスショウ

SIS(戦略情報システム)の文字が目につくビジネスショウ。文具類は電子文具に代わり、オフィスの電子化はまだまだ進みそう。



- ①これがAll in Note
- ②ファクシミリもフルカラーの時代
- ③電子手帳の住所録をハガキに印字する専用プリンタ
- ④ショウでは必ずこういうものもある
- ⑤ずらり並んだICカード
- ⑥Macintosh II でビデオフロッピーを読む
- ⑦いち早く Mach と国際化MOTIFを採用したLUNA88K
- ⑧大きく見えるが本当は小さいラップトップLUNA
- ⑨これがラップトップNEWSだ
- ⑩X-Windowでテレビが映る
- ⑪噂のバームトップ。これも68000マシンだ
- ⑫8cmCD-ROMのデータディスクマン
- ⑬XEROXの両面レーザープリンタ
- ⑭AXにつながったDVI
- ⑮これでオフィスから煙草の煙を追放できるかな?



コンピュータ関係を見るとラップトップパソコンは当たり前。今年は実用レベルでのカラー化ほか、各社とも A4 以下のブックパソコンを展覧して注目を集めていた。

そんななかでTFTカラー液晶を使ったシャープのラップトップAXの鮮明さ、そして8階調液晶と12MHzの80C286を擁し、さらに20Mバイトのハードディスクを内蔵して重さ2kg以下というAXパソコン、All in Note (MZ-8376A) の小ささが目をひく。

特にAll in Note。大きさはA4よりひとまわり小さいフリートップサイズで、Dyna Book比60%以下の容積で収まっている。補助バッテリーユニットをつけると奥行きが少し大きくなるが、その場合は5時間の連続使用が可能となる。

もっとも小さく、軽く、大容量で性能的

にもPC-9801NSに次ぎ、しかも大画面。

家電も含めてシャープは昔から製品を小さくすることでは定評があるが、とにかくこれで、やっと本当に使えるブックパソコンが出てくれたという感じだ。

さらに、ワークステーションもラップトップ時代に突入。デスクトップサイズになって驚いていたのが嘘のような勢いだ。小さくても内容はパソコンとは一線を画す。メインメモリは当然メガ単位、液晶やプラズマの高精細ディスプレイ、100Mバイト単位のハードディスクを内蔵する。ただ、用途はいまいちはっきりしない。

マルチメディア関係ではCD-ROM が元氣。実績で先行する富士通&ソニーのCD-ROM XAとインテルの推す DVI などが展覧。CDIはみつけれなかった。画像圧縮

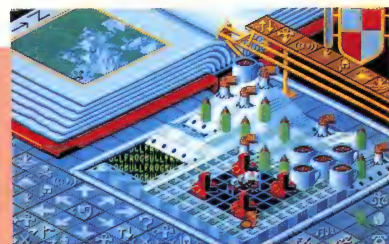
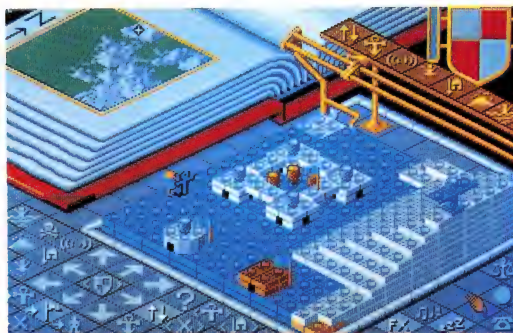
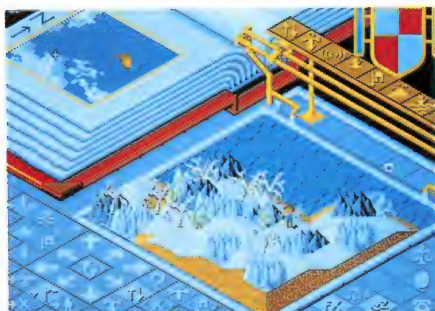
では一歩先を行くDVIはCD-ROM1枚に70分以上の動画像を収録する。デスクトップビデオの夜明けは近いかな?

そのほか、ソニーはデータディスクマンを展示。これはCDシングルサイズの CD-ROM専用端末だ。標準装備のCD-ROM 1枚に数種の英和辞典と和英辞典、国語辞典、漢和辞典などをまとめて収録している。これだけでも十分実用になるが、もっとも可能性は秘めている分野だけに、今後の展開が注目される。ディスクマンだから、もちろんCDシングルの音楽も聞けるぞ。

さて、今回のショウでは会場に女性の姿が目立ったのが印象的だった(無論、コンパニオンではなく)。嬌声をあげて大江千里のパンフレットを掻き集めるOL。ビジネスショウも変わった……。 (S.N.)

SOFTWARE INFORMATION

さて、今月もいろいろなソフトが出揃いました。すでに発売されているものもあるので、雨の中いまから買いに走ってあとは家にずーっとこもるといふ大技に出るのもいいかも(ただし時間が許せばだけど)。

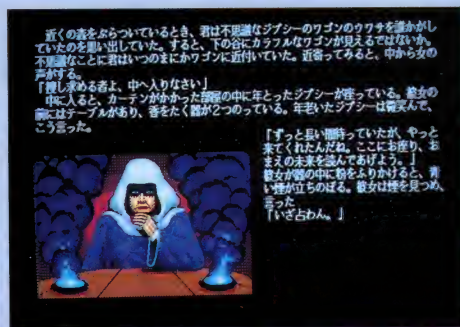


プロミストランド

あのポピュラスのシナリオディスクです。フランス革命、江戸時代などの歴史に沿ったゲームが展開できます。

ウルティマV

いわゆる悪者と戦うことから離れ、人間の深層心理を舞台にしたRPG。心のなかの正義と悪との葛藤を描いている。7月21日発売予定。



話題のソフトウェア

あー、うっとうしい! 何がうっとうしいかって? 決まってるでしょ、雨よ雨、梅雨の雨。まったく、この雨のおかげで靴はぐちゃぐちゃになるし、傘は持たなきゃならないし、持病の喘息発作は起こるしで、いいことなんかありやしない! あーあ、早くこの雨やまないかなー(なんて書いて梅雨になってなかったらどーしよー、何を隠そういまはまだ5月なのだし……)。

さて、この雨を蹴散らすかのように、今月もたくさんの新作情報が集まりました。

まずはこの**プロミストランド**からいきましょう。いまや人気絶頂のリアルタイムシミュレーション、ポピュラス。つい先日、編集部でも2時間15分にわたる西川善司VS

祝一平の対戦が行われたほど。まあ、この結果はまた後日お伝えするとして、今度はそのポピュラスのシナリオが登場するわけです。江戸屋敷ステージほか5つの舞台が設定されています。これでまた飽きることなく、新たな楽しみが味わえそうですね。発売はイマジニアより7月6日の予定。

お次はポニーキャニオンから発売の**ウルティマV**。お馴染みアメリカで大ヒットしているRPG、ウルティマの最新作です。ずっとX1でシリーズが出されていたので一瞬喜んでしまった方、ごめんなさい。今回はX68000での発売なんです。で、今回の諸悪の根源は、なんと心の影の化身シャドー・ロード。そうです、今回は人の心の「正義と悪」がテーマなんです。何が正しくて何が悪なのか、それを探し求めて再び聖者アバターはブリタニアを旅する、といったストーリー。操作性もグンとよくなったし、

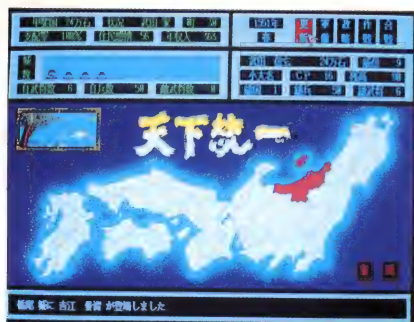
ついにポピュラス王座へ!

- | | |
|-------------------|----|
| 1. ポピュラス | 2 |
| 2. ダンジョンマスター | 1 |
| 3. ワンダラーズ・フロム・イース | 3 |
| 4. グラナダ | — |
| 5. 三国志II | 12 |
| 6. サーク | — |
| 7. ソーサリアン | 7 |
| 8. アルガーナ | 5 |
| 9. ファーストクイーン | 8 |
| 10. ジェノサイド | 6 |
- ポピュラスが着実に票を伸ばし、トップの座

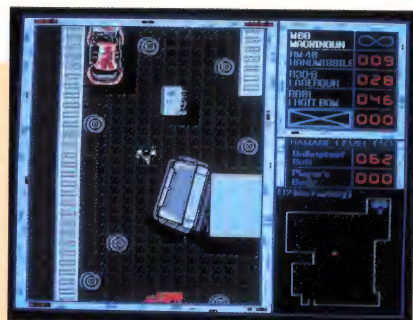
を奪いました。長く遊べるタイプのゲームは一旦トップを奪うとしつこいですから、しばらく居座るんじゃないかな。夏には、新シナリオを携えたダンジョンマスターを「プロミストランド」で迎えることになりそうです。

イースファンの皆さんにはすまませんが、今月も「ワンダラーズ〜」は3位でした。「ファルコム」にソフトを出させるために、みんなで買おう! 目標5万本!」なんてハガキもあって、1枚1枚に相当力が入っているんですけどね。でも、イースI・IIの発売は〇〇からなんですよ。ナイショだよ。

チャートのまんなかを見ると、グラナダとサ



天下統一



REINFORCER



実戦ビリヤード



ユニオン



ジェミニウイング

なかなか考えさせてくれて楽しめそう。

そしてシステムソフトからは**天下統一**がいよいよ発売となりました。本格派の戦国シミュレーションですが、難しく考える必要はまったくナシ。ただ、日本史に興味があれば、より楽しめるでしょう。さあ、みんなで天下統一を目指そう。

次々と新作を出してくれているザイン・ソフトでは、ただいま**REINFORCER**というシューティングゲームを開発中。詳しいことはまだお伝えできないけど、とりあえず画面写真が届いたのでお見せしちやいましょう。このほかザイン・ソフトでは、グラフィックツール**Gツール**も開発中です。

さて、システムサコムの**ジェミニウイング**ですが、着々と進行しているもようです。とりあえずデモソフトが届きましたのでお見せしますね。発売は7月中旬の予定です。

パック・イン・ビデオでは、**実戦ビリヤード**を移植開発中とのこと。これはPC-9801で人気だった同ゲームをさらにバージョン

アップしたもの。ナインボールのほか、ローテーション、四つ玉、スヌーカーなども楽しめる本格派です。

新規参入のポニーテールソフト/STUDIO オフサイドでは、麻雀牌などを使ったパズルゲーム**ユニオン**を開発中。これはステージ上におかれている牌を、同じものどうしで合体させていくというもの。牌は動物、くだもの、昆虫などがあって好みで選べます。発売は6月末から7月上旬の予定。

さてさて、タケルソフトからは**びんびん麻雀ピーチエンゼル**が発売中。お相手を務

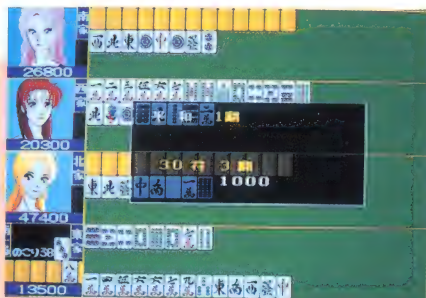
めるのはもちろん美女。お楽しみが見られるかどうかは腕次第ってとこかな。

スタジオパンサーからは**天九牌Special 桃源の宴Part 2 女子高生編**がタケルより発売中です。これは、天九牌のなかの桃源境編を抜き出し、ボーナスステージや女の子を増やして強化したものです。

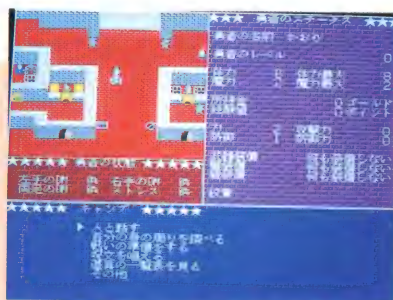
タケルといえば、LOGINのソフトウェアコンテストの優秀作が3本発売されています。RPG**風神魔伝II**とシューティングゲーム**PLANET**、パズルゲーム**フェブリー**がそれ。2,000円ならお買い得ってカンジです。

そうそう、光栄からX1turbo用シミュレーション**大航海時代**が出ました。世界を回り最高爵位を目指すゲームです。

そのほか、レイトレーシングソフト**C-TRACE**がバージョンアップされて発売されます。計算時間のスピードアップ、ツリー構造の採用などさらに使いやすくなって登場です。6月下旬発売予定です。では、来月。



びんびん麻雀ピーチエンゼル



風神魔伝II



PLANET



天九牌Special 桃源の宴Part 2

ークの新鋭が揃っています。サンプルから製品になって見違えたグラナダはまさに追ひこみの勝利。サークはビジュアルへのこだわりが、X68000ユーザーの胸に響いたようです。これ以上ランクアップできるかどうかは微妙なところですが。

この2つに挟まれて三国志IIも返り咲き。なんてったってX68000にもない新作。X1ユーザーのみんな、一緒にいぼう。やーいやーい。それにしても、X1用が3作もランクインしてるあたり、X1ユーザーのパワーも馬鹿になりません。

さて、来月グラナダは三強の一角を崩せるか？X1ユーザーの意地、三国志IIの動向は？(浦)

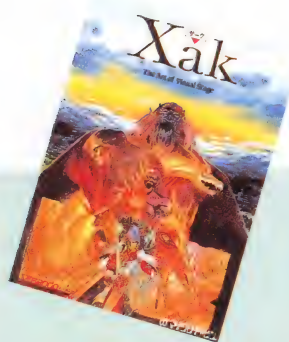
●Xak



与Xakは木を切るか

Nishikawa Zenji
西川 善司

他機種ではもうおなじみの「Xak」がついにわれらのX68000にも登場。これを紹介してくれるのはイースIIIの冒険記を終えたばかりの西川くん。マドルの次は与Xakの冒険だそうです。……もういいって。



X68000用 5"2HD版 4枚組 8,800円(税別)
マイクロキャビン ☎0593(51)6482

私が水爆頭の西川善司です。突然ですが私は3DダンジョンタイプのRPGが苦手です。XTALソフトの「ファンタジアン」は最後までプレイしたけれど、それ以後はまったくこのタイプのゲームに魅力を感じなくなってしまうのです(ファンタジアンは面白かったなあ)。私はそもそもあのマッピングというのが大嫌いでありまして、まして「すり抜けの壁」やら「ワープ」「回転床」などが出てきてしまうと、もういっぺんにやる気が失ってしまうのです。

あと、このタイプのゲームに多い緊迫感のない戦闘シーンが私は嫌いです。あの「1. 戦う 2. 逃げる 3. 魔法」というメニュー式のやつですよ。これってゲームをしているという感じがしないんですよ。

そんなわけで、私は大人気のビクター音産の「ダンジョンマスター」も食わず嫌いでいまだにプレイしていません。それに対して、この「Xak」のようなアクティブ・ロールプレイング(ARPG)は取っ付きやすくて面倒なマッピングもないし、きわめて私のような人種向きのゲームといえます。

さて、この「Xak」は去年の夏に発売されたPC-8801版がオリジナルです。思えば去年は各ソフト会社がARPGをこぞって出した年でありまして、ファルコムは「イースIII」、システムサコムは「プロヴィデンス」「ヴァルナ」、エニックスは「オールド・ビレッジ・ストーリー」、テクノソフトは「新九玉伝」、……と挙げればきりがなくらいです。ただ、ストーリーが「高貴な血筋を引く主人公が、魔王の復活を阻止するために立ち上がるが一足遅いで復活してしまう。が、x個の聖なるアイテムで封印する」というパターンが多く、先月レビューしたファルコムの「イースIII」もこの例にもれません。

本当に私は神の子孫なのか

俺の名は与Xak。まだ気立ての良い女房はいないが、木を切るのが好きな好青年だ。北島三郎の歌とはこれっぽっちも関係ないのは先にいっておこう。

エリス「おじいちゃんがメガネをなくして困っているの。探すのを手伝ってくれない?」

与Xak「あ、いいけど」

このエリスっていうのは町長の孫娘で俺のスケッ、じゃなかった幼なじみだ。

エリス「じゃあ、教会を見てきてくれないかしら」

うおおおお。ガンガンガンガン(頭をキーボードに打ちつけている音)いきなり

初めっから使っただけ。こりや、先が思いやられるなあ。で、眼鏡を持っていくと……。

町長「おお、待ってあったぞ。お前の家を訪ねてはるばる南のお城からお客が来とるんじや」

メッセンジャー・ピクシー「あなたが戦士ドルクさんですか」

与Xak「ドルクは俺の親父だよ。親父は半年前に旅に出たきり行方がわからないんだ」

メッセンジャー・ピクシー「そうですか。戦いの神デュエルのいない今となっては子孫であるあなただけが頼りなのです(きた、きた、きたぞ〜)」

与Xak「こ、この俺が神の子孫! うおおおお、ガンガンガンガンガン」

メッセンジャー・ピクシー「どうかしました?」

与Xak「神の子孫が、眼鏡探しをさせられたなんてあんまりだと思って……」

ARPGの主人公は風呂に入るのか

武器を買い揃えたら町の外へ出よう。外にはスライムやスケルトンなどがいる。戦闘の方法は単純明快。「イース」や「ハイドライド」と同様、剣を構えて敵に向かって突っ込んでいけばいいのだ。ただし、真正面から突っ込んでいったのではダメージを受けやすいようだ。敵の背中、または側面を狙うようにしたほうがいだろう。

また、ダメージを食ったら敵の来ない安全な場所で休もう。全角キーをONにしゲームスピードを上げてから休むと回復のスピードも速くなって気持ちいいぞ。

細い山道を通って東のほうへ進むと、「西遊記」の八戒のようなやつ(オーク)と赤い手長猿のような敵(トロール)の出現する場所に出る。レベル2になったらここで修業するといだろう。

さて、その北には穴の中でゴーゴー(死語)を踊るワーウルフの森がある。レベル3になったらここがよい経験値稼ぎの場所となる。おや、あそこで寝そべっているの



おや、こんなところに女の子が倒れているぞ

は？ ……女の子だ。大変だ、ケガをして倒れてるぞ。

与Xak「ど、どうしました？」

フレイ「ケガをしてしまったの。家が近くにあるから連れていってください」

与Xak「よし、背中に乗りなさい」

フレイ「……くさい」

与Xak「ははは、そりゃそうだ。冒険に出るから一度も風呂に入っていないからね。『イース』シリーズの主人公アドルは3年入っていないそうですぞ。きっと奴の体はタムシと水虫だらけだろうな。西川善司でさえ、1週間に2回は入るのにねえ」

フレイの家は彼女が倒れていた場所から東へ行ったところにあった。

フレイの父「おお、フレイ。無事だったか。あれほど水汲みはいいといったのに。これはひどい臭いと傷だ。与Xakさん、娘を町の病院まで連れていってくれませんか？」

与Xak「うおおおお、ガンガンガン」(XF3キーがとれちゃった)

フレイの父「どうなされました？」

与Xak「あ、ただの発作です」

病院の看護婦はオレンジロードか

病院はフェアレスの町の南東の隅にある。

看護婦「あら、与Xak君。……どうしたの、その女の子は？」

与Xak「森でモンスターに襲われてケガをして倒れていたのを俺が助けたのさ。診てやってくれないかな。それよりあんな、『きまぐれオレンジロード』の鮎川まどかにそっくりだね」

看護婦「うふふ。MSX版の『Xak』の看護婦なんか『めぞん一刻』の管理人さんの音無響子そっくりだったんだからあ」

与Xak「はあ？ じゃあ、FM-TOWNS版はちびまるこちゃんかサザエさんのワカメちゃんだなあ、きっと」

看護婦&与Xak「ははははは」

……マイクロキャビンさん、こういう同人誌レベルのギャグは真面目なゲームに入れると逆効果だと思います。



森の奥にどっしりとかまえている木の精

木の精はレベルいくつで倒せるか

フレイを病院へ送り届けたことを親父さんに告げると、今度は森の木の精を倒してほしいという。木の精はワーウルフの森のいちばん奥にいる。が、レベルが7以上ないとまったく歯が立たない。そこで以下に各レベルで倒せるモンスターのうちでいちばん経験値をくれるやつを記しておこう。レベル11以降は自分で調べてちょうだい。

レベル4 バジリスク(トカゲみたいな奴)

レベル5 トレント(木の化け物)

レベル6 サイクロプス(一つ目の巨人)

レベル7 毒沼に住む赤い奴、サンドマン、

Dファイター(砦にいる騎士)

レベル8 木の精

レベル9 ケルベロス(砦にいる犬)

レベル10水竜(砦のボス)

これを見てもわかるように、木の精を倒したあとに行くべきところは「砦」だ。先は長いぞ、がんばれ。

Xak2は出るか出ないか

「Xak」はなかなかよく出来ていますが先月レビューした「イースIII」と比べてしまうと、ちょっとムムムの点があります。まず、スクロールがX68000なのに16ドット単位だということ。オリジナルのPC-8801版が、キャラと背景の重ね合わせをチップ単



まさか、この犬は火を吐かないだろうな

位で行っていたからといって、そのまんな移植することはなかったと思います。

そして、やっぱりストーリーに新鮮味がないですね。結局最初にいったARPGのストーリーパターンそのまんな。大筋はそうになってしかたないとしても「イース」シリーズのようなドラマチックさがもっと欲しいところですよ。1つひとつのイベントに登場するキャラクターが使い捨て的に出てくるだけで、ストーリー全体を通しての伏線が張られていないのです。キャラクターが単なる「主人公にアイテムを渡す人」なんですね。だいたい主な登場人物と紹介されているヒロインのエリスとフレイがゲーム前半に出てきただけで、そのあと一度も出てこないのはどういうわけでしょう。

最後にゲームバランスについて。この「Xak」では、経験値稼ぎをさせられる局面によくぶつかります。新しいフィールドに移ると、そこに出現する敵にまったくダメージを与えられず死んでしまうことがよくあるのです。ボスキャラも同様です。そうすると前のフィールドに戻って弱い敵をレベルアップのためにえんえんと殺し続けなくてはいけないのです。この辺も「イース」シリーズを見習ってほしいところですよ。

おまけ：ゲームの最初のピクシーの頼みを断り続けたり、フレイをおぶったままエリスの家に行くとエリスの表情が変わる。

総 評

ソーヒョー！(釣りキチ三平のまね)

いいたいことは本文に書いたもので、ここではBGMとグラフィックについて書きたいと思います。

「Xak」はBGMがとてつもないです。リズムはAD PCMでやっているし、鶏の鳴き声なんかのSEが入っちゃったりベースのソロなんかも入ったりして、ゲームのBGMとしてはとても斬新な感じがしました。ゲームミュージックファンは要チェックです。

一方グラフィックですが、店やイベントで表示される人物などの絵が最近ありがちなオタクキーなものとは一味違ったタッチで、色の使い

方も綺麗で良いと思います(EXCEPT 病院の絵)。しかし、通常のゲーム画面の背景はいかにもチップで構成しましたという感じで「イース」なんかと比べると多少劣ります。キャラクターもなんとなく寸詰まりです。この辺はもう工夫するべきだったと思います。

ソーヒョー！(釣りキチ三平のまね)

グラフィック	7
BGM	9
ゲームバランス	5
ストーリー	6
操作性	7
ディスク周り	6
技術	6
総合点	7



そんな顔しないでよ、看護婦さん

●あーくしゅ



ウルフファン必須の パロディゲーム

Koga Kazunori

高河 和海苔

パソコンゲーム界に数々の名作を残してきたウルフ・チームが放つ、ファン必見のパロディアドベンチャーゲーム。いままでのゲームの主人公たちがかわいくデフォルメされて、画面のなかで大活躍するぞ。



X68000用 5"2HD版3枚組 6,800円(税別)
ウルフ・チーム ☎03(5273)4795

ドンドンドンドン、と無言のノック。この春もまた新聞勧誘員のせいで、とても不愉快な思いをしました。

ドアを開けると、靴の先をドアの隙間に差し込み、不当景品をちらつかせながら、「いまだこの新聞をとってるの?」と、いつものセリフ。知ってるくせに! ま、その程度ならまだしも、「俺を誰だと思ってるんだ。玄関のドアがなくなってもいいのか」だとか「そこのフライパンで殴ってやろうか」だとか言われたこともありました。その夜からは数日間おびえながら生活していた小心者の私です。

思いあまって某大新聞社に直訴の手紙を出したこともあったのですが、「そんな末端の事情は知ったこっちゃない」ってお返事でした。ああそうですかあ。

とにかくかわいい〜

そんな憂鬱な気分も和らぐ「あーくしゅ」を紹介しましょう。ハッキリ言って、遊んでる間ずーっと面白かったゲームというのはひさしぶりです。ニュージーランドストーリーに次ぐ、私のおすすめ品にしましょう。しかし、独創的なシステムだとか、きわだって優れたシナリオだとか、そういう面からの評価はできません。普通のコマンド選択式アドベンチャーであり、アイテムを集めて問題を解決させるんだタイプのシナリオだからです。じゃ、何がよかったのかというと、グラフィックと会話、ほのぼのした雰囲気が気に入ったのです。とにかく、かわいい。3等身にデフォルメされたキャラクターたちがとってもかわいい。じえだやスーニーの大ボケな言動がかわいい。サーラのムチがいとおしいってなもんです。

まあ、「買い—捨て」の判断は人によってまちまちで「出来がいい—悪い」だったり「メジャー—マイナー」だったりするかもしれませんが、なにがなんでも無条件におすすめとまでは言いませんが、「かわい—かわいくない」を評価の基準に据えている方々には、ぜひ楽しんでもらいたい作品だな、と思います。

パロディなんです

「アークス」を舌つらずに発音したタイトルからも想像できるとおり、あーくしゅはアークス、およびアークスIIのパロディになっています。

そのほかにも、ちまちまとした戦闘の「ファイナルゾーン」や、渋さが売りものの「ヤシャ」、優子と麗子の「夢幻戦士ヴァリス」、いまひとつ知名度の低い「ガウディ」、壮

大なストーリーの「ミッドガルツ」、凝ったオープニングが評判の「斬」、最新作「グラナダ」、といったゲームからも、キャラクターたちが参加しているという、いわばウルフ・チームから生まれたキャラの大パーティーなんです。

そんなわけで、元ネタを知っているほうがより楽しめるあーくしゅですけど、知らなくても大丈夫です。少なくとも知ってなければ困る、解けないなんてことはありませんので、安心してプレイしてください。

聖剣と仲間を探すのだ

さて、主人公はアークスから引っ張ってきた、じえだとピクトです。この2人が一緒に冒険をするわけですが、なぜ2人組なのかっていうと、じえだだけだと話が進まないし、ピクトだけだと面白くないからなんでしょうね、きっと。ピクトが常識をわきまえた好青年なのに対して、じえだは脳天気の大ボケ野郎で、なにかといえば「おなかすいたね〜」「おべんと〜」「ほう」なんて言ってる無邪気な22歳なんですから。まったく、かわいいぞ、こいつう。

それではゲームスタート。まずはどこかの田舎道からです。2人はぼんやりと旅を続けていたようですが、少し歩くと金竜リグ・ヴェーダに出会い、時間の秩序を守るために協力してくれ、と頼まれます。

いきさつを簡単にお話しましょう。むかしむかしから、3本の聖剣というものが存在していて、これらがすべての時と空間を支配していました。ところが、いろいろな時代を行き来していたものですから、いくつかの次元に裂け目ができてしまい、世界が混じり始めてしまったのです。このままでは、時間の流れが乱れてしまいます。この裂け目を埋めるためには、3本の聖剣を合わせたエネルギーが必要です。裂け目を通していろんな時代に行き、聖剣を集めてください。3本の聖剣はそれぞれ違った音楽に反応してその姿を現すので、まずは3枚のCDを探してください。と同



冒険のはじまりはじまり〜

時に、裂け目から吸い込まれてしまった仲間たちもこの時代に連れて帰りましょう。では、よろしく。ってな具合です。

この役目を請け負った2人は、次元の裂け目に突入、そこにはなぜか、U子とO子（優子と麗子ね）が店員をしているDOSバーガーがあるのでした。

O子 「いらっしやいませ、こんにちは」
じえだ 「はんばあぐ」

O子 「はい？」
じえだ 「ぼてとお」

O子 「あ、すみませんもう一度……」
じえだ 「こおら」

O子 「あ、あのお……」
じえだ 「ぴ〜くと〜お、か〜えな〜いよ〜お」

まったく、じえだはまともに会話もできないんだから。ほかにルアン・カーンっていう、どこにでも出てくる変なのもいます。
ピクト 「すごい雲の波だねー」

じえだ 「すご〜い」

ルアン 「ははははは！ 愛と勇気の男ルアン・カーン参上！」

じえだ 「すご〜い、くものうえになってる〜」

ルアン 「見よ！ この勇姿！ すべての女性を魅了する甘いマスクと強靱な肉体を兼ね備えた光の騎士ルアン・カーン！ ふはは、世界は私のためにあーり！ ふはははははははは！ わあー！」

ピクト 「落ちた……」

そんなこんなで、くだらないじゃれが終始ちりばめられつつ冒険はなごやかに行われます。で、もちろん超未来、近未来、現代、過去を駆け回って、3本の聖剣と6人の仲間（エリン、グラン、スーニー、バザン、サーラ、チノップ）を探すという本来の目的も大切ですが、このゲームではそこまでの過程、1つひとつの会話もじっくり楽しんでほしいと思います。

ちなみに、エンディングはアニメ番組のそれっぽくて、いかにもふさわしい感じのものでした。

親切な設計

最近ゲームオーバーのないアドベンチャーゲームやロールプレイングゲームが増えています。このあーくしゅも一応ゲームオーバーがあるのですが、実はないのであって変わりません。

それはガウディにもあった、ヒストリーリピートシステムの搭載によります。ゲーム中であれ、画面上にあるHRっていうアイコンをクリックすると、過去20画面分が縮小されてずらずらっと表示されるのです。この中からどれかをクリックするだけで、そこからやり直すことができます。これは便利。もちろん普通のセーブ機能もあります。

それから細かいことですが、マウスカーソルの形が矢印や指ではなくて、セーラー服の女の子なのが泣かせます。こうした遊び心は大歓迎ですね。

おまけに、ミュージックモードもついていて、音楽を聴くだけでなく効果音を聞いたり、グラフィックを見たり、スタッフからのメッセージを読んだりもできます。BGMはこれまでのゲームをアレンジしたのですが、音色には相変わらずのパラメータを使っているようです。

フルマウスオペレーションといい、十分な処理速度といい、完成度の高いシステムに仕上がっているといえるでしょう。

しかし、多くのアドベンチャーと同様に、Aから手にいれたBをCに渡すとDをくれるので、それをEに渡すとFに案内してくれて、その場所でGで買ったHを使うとIとJとKが見つかるので云々、と不条理なつながりで、ストーリーが進行していく事実は否めません。パズルゲームのようで、面白いといえば面白いんですけどね。

最後にヒント

あーくしゅはすでに市販されているので、



江戸時代までであるとは



ミュージックモードもまた楽し

もうとっくに解いちゃったよ、っていう人もいることでしょう。が、このテのゲームに慣れてなくて、ハマってしまっている人もいるかもしれません。そんなわけで、ちよっとヒントでも書いておきましょうか。

だけど、これから遊ぼうと思っている人は以下の文をゼ〜〜〜ツタイに見ないように。ではでは、がんばろう！ ほう！

- お金を貯めるには、超未来に出てくる敵Mの顔を誉めるのが手っとり早い。
- 眠っているEには、彼女の好きなものをさらに飲ませてあげよう。
- Sはムチの置いてある店にいますので、しつこく探してみよう。
- 光の浪人Rなんかに大切なモノを渡したくはないが、見返りが期待できる。
- 過去はしばしばろつてみることに。
- 封印のされた洞窟を開けるためには、Sを同伴していなければならない。

<総評>

ゲームの購入の選択は、お金の有無なしと、時間のあるなしによって、結構変わってくるものかもしれません。

時間があるのにお金がない人は、光栄の歴史3部作のようなSLGなどを買って、じっくりと楽しみたいと思うのではないのでしょうか。また、時間がなくてお金がある人は、テトリスやアフターバーナーなどを買いそろえて、気分転換にちょっと遊んで、なんてぜいたくをしていることでしょう。

あーくしゅは、日曜のお昼に始めたら、夕方

には終わってしまいそうなお手軽アドベンチャーです。特にこんな方にはぴったりだと思います。

なお、時間があるうえにお金もあるお方、および時間もないくせに金もない！ って方に対するコメントは控えさせていただきます。礼。

評価表（最高は♥10個）

グラフィック	♥♥♥♥♥♥♥♥♥♥
シナリオ	♥♥♥♥♥♥♥♥
サウンド	♥♥♥♥♥♥♥♥
完成度	♥♥♥♥♥♥♥♥♥♥
少女趣味	♥♥♥♥♥♥♥♥♥♥
ほのぼの	♥♥♥♥♥♥♥♥♥♥



次元の裂け目を通ると、そこは……

●ダウタウン熱血物語



ストレス解消用 ケンカアクション

Yamada Junji

山田 純二

ファミコンからの移植でおなじみ「熱血硬派くにおくん」シリーズ第3弾。今回はくにおくんのライバル、りきが主役って感じです。さてさて、いったい何が起こることやら……。



ファミコンで人気のダウタウン熱血物語が、シャープから発売になりました。お馴染みSPSによる移植ですが、移植の完成度はもちろん完璧、元のゲームの味をそこなく、元気たっぷりのアクションが楽しめます。

ゲーム内容は、群がる不良どもをバツバツ薙ぎ倒し、相手から巻き上げたお金を使い、商店街で暴飲暴食をして（なんてヤツだあ!）、どんどん強くなっていくという、ストレス解消にはもってこいのアクションゲームです。とはいえ失敗して「ふくろだたきの刑」にあうと、逆にストレスが溜まりますけど。

このゲームでは、熱血高校ドッジボール部でお馴染みのくにおくんともうひとり、くにおの永遠のライバルである花園高校番長りきの2人によって、ゲームが進行していきます。ひとりでも2人でもプレイは可能で、2人プレイではくにおとりきが互いに反発しあったり、協力したりという2つのモードが用意されています。仲のよい友達とプレイする場合には協力モードで、嫌な野郎とは反発モードで（でもわざわざやなヤツとはやらないか）遊んでみるといいかも。ちなみに、この2つのモードの違いは、プレイヤー同士の当たり判定があるかないかだけです。

ひとりの場合は、りきかくにお、どちらかを選んでプレイするんですが、ストーリーを読んでみると、今回の主役はどうやらりきのよう。くにおくんは単なるオマケのような感じで、くにおくんが好きな私は、ちょっと寂しい……などとぼやいていてもしょうがない、そろそろりきのお手伝いでもしに行きましょうか。

運命の再会

花園高校、校門前。怒りながら走ってくる人影と、校門前に立っている人影が見える。

くにお「りき、どこへ行くつもりなんだ」

りき「く、くにお！ おまえ、なににし……」

くにお「冷峰学園へ行くんだろう。ちょっとつきあわせてもらうぜ」

りき「きさま、どうしてそれを知っているんだ！ まさか、今回の事件に、お前も絡んでいるんじゃないだろうな」

くにお「残念ながら、関係ないな。どうした、行くのか、行か

ないのか」

りき「勝手にしろ！」

今回はなんと一大事！ 最近ハバをきかせ始めた冷峰学園の生徒に、りきの彼女がさらわれてしまったのだ。そして、りきの手元に届いた1通の挑戦状……。「彼女を返してほしければ、冷峰へこい」。

こうして、2人は冷峰学園へ向かったのでした。

冷峰学園ってどこ

りき「ところで、おまえ冷峰学園がどこにあるか知ってるか」

くにお「……まさか、おまえ知らないって言うんじゃないだろうな」

りき「そりゃあ、挑戦状に書いてなかったし、あんな弱小学園、いまだにかけたことなんかないからな」

くにお「おめでたい野郎だぜ……」

なんとも間抜けなお話、挑戦状を受け取った方がいいが、2人ともかんじんの冷峰学園の場所を知らなかったのです。このあと何度か罵りあった2人、結局は地道に冷峰を探すことにあいました。

さて、冗談はともかく、マップの繋がりはかなりややこしく、空間が歪んでいるとしか思えないほど（!）の場所があるので、自分の方向感覚に自信がない人は、マッピングしていきましょう。ちなみに私は、ちょうど暇だった編集のA氏をひきずりこんでマッピングをやらしてしまいました（おいおい）。

ゲームを始めて、まずたどりつくのが花園商店街。そこを抜けていくと花園第三公園へたどりつきます。ここらへんから、周辺の高校生の不良どもが現れるので、ようやくゲームらしくなってきます。

くにお「おい、坂宿高校のやつらがやってきたぜ」

りき「ふん、あんなやつらなんかどうって



うっ、目が飛び出るほどの木刀の痛さ！

ことないさ」

くにお「強がってんじゃねえよ」

りき「なんだとこの野郎！」

と言いつ争っている2人を襲う鉄パイプ2本。スコーン！

くにお「いってー、あんの野郎よくもやってくれたな、ただじゃおかねえ」

りき「俺様の實力を思いしらしてやる。かくごしやがれ！」

んー、まさに熱血、単細胞。まったく、頭に血が上ると見境なくなるんだから、この2人は。てなわけ、アクションゲームの本領発揮です。このゲームでは、手に取れるものは全部武器となります。たとえば、敵さんが残していった木刀、ナックル、チェーン、そして鉄パイプ、さらには道端に転がっているポリバケツ、タイヤ、石ころ、古本、きわめつけは気絶して倒れている人間さえも、持ち上げてぶん投げる事ができます。特に、チェーンを振り回しているときなんて、気分はもう「おーほっほほ、女王様とお呼び！」の世界、うーんチェーンの唸る音がとても気持ちいい（あつぶねエ）。

ところかわって桜町

なにやら薄汚い工事現場。敵を倒しそのまま進もうとしたら、どこからともなくディスクアクセスの音がして沢口君の登場。沢口「ちょっと待ちな！俺はおまえらのようなやつらを通すなと、西村さんに言われている。ここを通りたかったら俺を倒すんだな」

くにお「俺たちは、てめえなんざにかまってるヒマはねえんだヨ、怪我しないうちに下っ端はとっととすっこんじまいな」

りき「そうそう、冷峰学園の場所を教えてください」

沢口「なっ、しゃらくせえことほざいてんじゃねえ！ぶっ殺してやる！」

などという会話があったかどうかは知らないけど、戦闘開始です。ここは足場が狭いので、工事現場に落ちこまないように注



くにおと長谷部、感動の再会!?

意。先手さえ取れば、たいした相手ではありません。マップのあちこちには、このように中ボスが存在します。なかには有益な情報をくれたりするやつもいるので、余裕があれば逃げずに戦いましょう。

しゃべるんです

ところ変わって、ここは新宝川花園大橋。くにおは中学時代の同級生、長谷部と再会することになる。

長谷部「ひさしぶりね、くにお♡♡♡♡。冷峰の長谷部さんが、とってもいいこと教えてあげる♡♡♡♡」

わあ、びっくりした。いきなりサンプリングでしゃべりだした長谷部さん。2行しゃべっては、ディスクアクセスしながら、えっちらおっちら話してくれた情報によると、りきの彼女をさらったのは、冷峰のダブルドラゴン竜一、竜二であることと、彼女は校舎のいちばん上のいちばん右の教室に、閉じ込められているらしい。そして、セリフの最後には、

長谷部「あたしは、冷峰の生徒だけど、くにおを応援するわ♡♡♡♡♡♡♡♡。絶対負けないでね♡♡♡♡♡♡。約束♡♡」

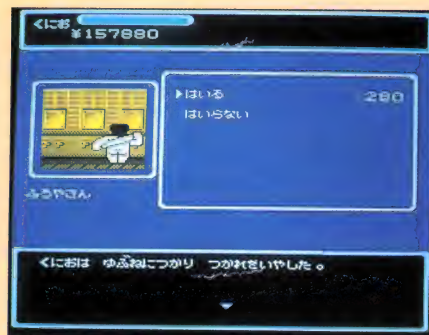
セリフの間のハートマークの連打。くにおくんも、あまりの迫力に圧倒されている様子。こころなしか顔が青ざめているようで……。

りき「たいへんだな、おまえも」

くにお「お互いさまだろ」

感想は……

このゲームでいちばん楽しいのは、食事のときのくにおくんのしぐさが(かわいい!)この一言につきます。もちろん、カッコいいアクションシーンも捨て難いが、くにおくんのいやいやにはかなわない、と僕は思います。ファミコンからの移植とあってゲーム性、バランス、ともにいいのですが、背景のグラフィックまでそのまんま使い回しが多いので、X68000でやっているとは結構違和感を感じます。効果音については、サンプリングを使っているだけあって、文句ありません。ただ、一言文句を言わせてもら



銭湯に入るとプリティなおシリが見られる



こんな軽ワザ師のようなマネもできちゃう

どうやら、りきのほうも彼女とは苦勞している様子。妙にしみじみとしてしまった2人、なにをやってるんだ! まだまだゲームを始めたばかりじゃないか、こんなところであげてる、冷峰へのみちのりは遠いぞ、がんばれくにお、そしてりき!

*

とまあ、こんな具合にゲームは進んでいきます(登場人物の台詞はちょっと違うけど)。このあと、りきとくにおは、緑町、白鷹町、星の丘町、そして夢見町を駆けめぐり冷峰へ到達しますが、冷峰学園の門は閉ざされていたのです。冷峰の四天王を倒さない限り、中には入れないようです。四天王のうち望月と平は倒した、残りの2人はどこにいるのだろう……。まだまだ先は長いようです。きしむジョイスティック、うなる必殺技、盛り上がるBGM、遊び心も一杯のこのゲーム、大作とはいいがたいが、とても熱くなれるゲームです。



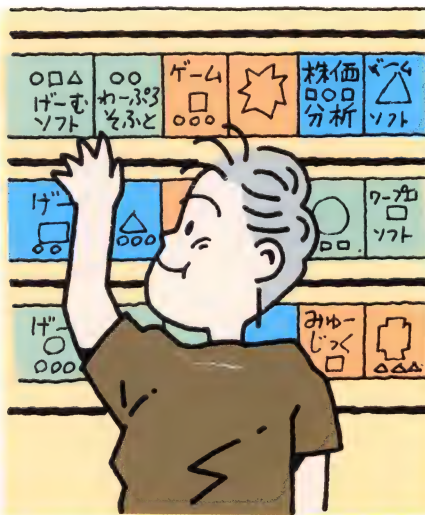
お店ではいろんなものを買えるのです

えるなら、背景についてもX68000を使っているんだから、もう少しなんとかしてほしかったなあ。この点だけが唯一残念だと思います。次にファミコンから移植するときは、ぜひともがんばってほしい。SPSさん、次作のワールドコート期待してまっせ。

グラフィック	7
操作性	9
効果音	10
アクション	9
買い物	8
熱中度	8

AFTER REVIEW

今月はグラナダ、FAR SIDE MOON, サーク, 三国志IIの4本を取り上げてみました。なかなか皆さんやり込んでいるらしく、実際にプレイした感想がひしひしと伝わってきます。こういったハガキがどんどん増えてくれると、うれしいですね。



グラナダ

- ▶とにかく全面ぬかりなく、面白さがつめ込まれて、感動することうけあい。演出がGood! 山形県・石澤 敏博(18)
- ▶サウンドにひどくひかれて買いました。でも、ゲーム内容はサイコー! アーケードをこえています! 静岡県・小澤 健一(16)
- ▶BGMと多彩なオプション、デカイボスなど、カッコよかったから。岡山県・加藤 雅浩(21)
- ▶ボスを見ると、オッと思わせてくれるから。長野県・小笠原 信夫(15)
- ▶爆発音が仕事の苦しみを忘れさせてくれる。東京都・山田 慎也(20)
- ▶難易度の変更できて誰にでも遊べる。神奈川県・光石 和弘(17)
- ▶地味だが長く遊べるよいゲーム。茨城県・石川 満章(22)
- ▶撃って撃って、撃ちまくりゲームだから。神奈川県・丸藤 俊之(21)
- ▶バランスがよく、サウンドもグラフィック



もよい。東京都・井上 正男(17)
▶2面の空のステージは、とくに面白い。もちろん落ちるとちーさくなってくれるし、4門の誘導ミサイルもとっていいと思う。が、ボスキャラには神風にならなければいけないのが、すごく悲しい。

香川県・大西 力(18)
▶グラフィックにメッセージを感じる? 変なビジュアルシーンがないのもよい。ただ、自機がチャチャイのが難。

三重県・長井 まさひと(24)
▶タンクゲームマニアがよだれをたらしてやりまくる気がするから。

茨城県・秋山 真一(16)
今回は、このグラナダに関するハガキがいちばん多かったですね。タンクタイプのゲームだけに、なんとなく古めかしいイメージを抱いてしまう人もいるかもしれない、なんてちょっと心配してしまったりしたのだけれど、そんな心配はまったく無用だったようで、皆さん楽しんでいるようです。なにを隠そう編集部でもこのグラナダは人気のマツで、来るたびに遊んでいくライターもいます。

最初は、操作性にちょっととまどうこともあるかもしれませんが、慣れてくると快適そのもの。ガンガン撃ちまくって気分も爽快。効果音も気分を盛り上げてくれますし、自分の腕に合わせてモードを設定できるので、誰でも楽しめます。しかし、まだMANIAモードでクリアした、という話を聞いたことがありません。お心あたりの方はぜひご一報を編集部まで。

X68000用 5"2HD版 4枚組 8,800円
ウルフ・チーム ☎03(5273)4795

発売中のソフト

★天下統一

黒田幸弘氏が手がけた本格派戦国シミュレーションゲーム。プレイヤーは、戦国時代の英雄となつて一城の主からスタートし、あるいは武将をしたがえ内政をほどこし、あるいは諸国の群雄を相手に野戦・攻城戦を繰り広げる。国を手中に治めたら、目指すは京、そして天下統一だ。今回移植を担当したのは、アルシスソフトだ。

X68000用 5"2HD版 2枚組 9,800円
システムソフト ☎092(752)3902

★天九牌Special桃源の宴Part2女子高生編

すでに発売されている「天九牌」から桃源郷編のみを抜きだしたのが「天九牌Special桃源の宴」。これはその桃源の宴シリーズの2作目。こんどは、女子高生9人がプレイヤーのお相手を務めます。女の子を破産に追いこんで、お金を貸してあげればさらに増えたグラフィックが楽しめますぞ。

X68000用 5"2HD版 2枚組 3,800円
ブラザー工業 (TAKERU) ☎052(824)2493

★風神魔伝II

LOGINソフトコン入選作のRPG。中世のたたずまいを持つフローランシア王国は、豊富な水と広大

な大地に恵まれた平和な国だったが、突然現れた風神が国々を荒しまわりはじめた。この風神を倒すため、主人公のキミは立ち上がった。操作性やグラフィックもなかなかよし。

X68000用 5"2HD版 2,000円
ブラザー工業 (TAKERU) ☎052(824)2493

★PLANET

LOGINソフトコン入選作。全7面のシューティングゲームだ。海、山、宇宙空間と豊富に設定されたステージを5種類のパワーアップを使って突き進む。各面の最後には巨大なボスキャラとの対決が待っている。

X68000用 5"2HD版 2,000円
ブラザー工業 (TAKERU) ☎052(824)2493

★フェブリー

これもLOGINソフトコン入選作。コースの上に置かれた鉄球を転がしてゴールへ導いてやればよいのだが、コースから落ちないようにするためには、磁石を置いてうまく誘導してやる必要がある。全50面の新感覚パズルゲームだ。

X68000用 5"2HD版 2,000円
ブラザー工業 (TAKERU) ☎052(824)2493

★大航海時代

中世を舞台に、7つの海を航海し最高爵位を目指すというシミュレーション。地中海からスター

FAR SIDE MOON

▶ややゆったりすぎる感じは隠せないが、ソフト開発etc.の流れがいい(6月号のP.37の『最後にひと言』には、共感した)。

群馬県・齊木 喜浩(16)

▶いろいろな戦略が立てられるところがいい。

北海道・河野 健司(15)

▶あのグラフィック、あの値段、あのソフトハウス、文句なしですハイ。

京都府・高 宏行(15)

▶作る楽しさを味わえる。

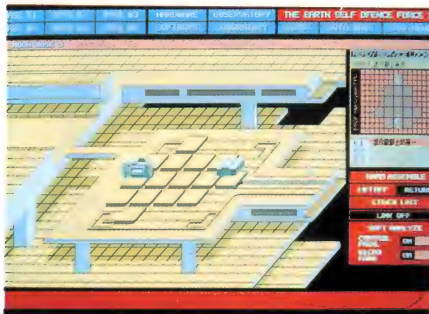
東京都 柴田 和久(17)

▶最近リアルタイム中毒のため。

埼玉県・筒井 圭一郎(17)

このゲームも純粋に楽しめるシミュレーションだったようで、例の件など気にもせずに皆さん遊んでくれているようです。なかなか“ものを作る”“頭を使う”といった点で好評でした。グラフィックもきれいでしたし、人気が高いのもうなずけますね。

X68000用 5"2HD版 3枚組 9,500円
アートディンク ☎0474(77)7541



サーク

▶移植に手を抜いてない(みたい)だから。

大分県・佐々木 亮(14)

▶PC-8801のときは、4MHzでしかできなかったから。OPMBには笑えた。POINTゲームは面白い。

千葉県・後藤 勝(21)

▶全体的に動きがいいしきれい。

神奈川県・池本 誠(19)

▶音楽、グラフィック、シナリオ、難易度がよい。

大阪府・平野 晃男(20)

▶イースシリーズと似ないための努力がヒシヒシと伝わってくるから!

神奈川県・石井 直樹(16)

▶最後のボスが強い、けど面白い。これは買いた。

静岡県・岡田 拓也(17)

比較的最近に発売されたゲームなのに、けっこうたくさんの方のハガキがきました。テンポよくプレイできるのが要因でしょう。そう難しいクエストもなく、とっつきやすいゲームといえます。

X68000用 5"2HD版 4枚組 8,800円
マイクロキャビン ☎0593(51)6482



三国志II

▶他国との外交関係など、いろいろな要素があって、とにかく面白い!(光栄さんのソフトの中ではいちばんいい)

山形県・水口 昌郁(21)

▶武力、知力ともにIよりも史実に近くなった。

滋賀県・田中 哲(17)

▶三国志IIは確かに難易度が低くなった。また、計略で忠誠度100の武将の引き抜きもできるようになったのはよい。ただ、200/400ライン兼用のためか漢字VRAMの使用を止めたことで反応速度が遅くなったのはおしい。

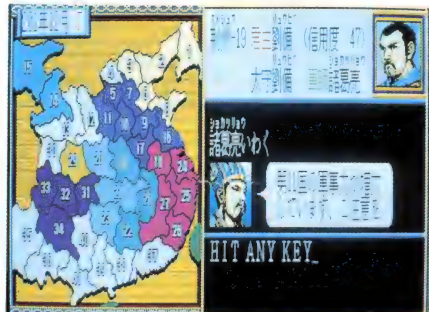
東京都・中北 康夫(23)

▶前作の欠点がいび改善されているし、MUSIC以外はよくできている。

岐阜県・西松 博章(19)

前作と比べてよくなったという方が多かったですね。かけひきや一騎打ちがよいという方も何人かいました。さすがはシミュレーションの光栄といったところでしょうか。

X1turbo用 5"2D版 3枚組 14,800円
光栄 ☎044(61)6861



トし、数々の港で仲間を集めつつ、貿易や戦闘、航路開拓などをして進めていく。

X1turbo用 5"2HD版 9,800円
光栄 ☎044(61)6861

新作情報

★ウルティマV

アメリカの大ヒット作、ウルティマの最新作がいよいよ登場。

ブリティッシュ王国が地下大陸に消えてしまっ以来、ブリタニアの国民は圧制に苦しめられていた。その根源は心の影の化身、シャドー・ロード。再び聖者アバターはブリタニアに降り立ち、人間の内的な正義と悪の葛藤の旅に出る。

X68000用 5"2HD版 2枚組 9,800円
ポニーキャニオン ☎03(221)3131

★ユニオン

X68000に新規参入のポニーテールソフト/STUDIO オフサイドによるパズルゲームだ。画面に置かれた牌同士を縦横自由に合体させ、巨大な牌にしようのが目的。キレイに書き込まれた牌は、麻雀・昆虫・果物・動物に自由にコンフィギュレーションできるから、皆さんの好みに合わせてどうぞ。

X68000用 5"2HD版 7,800円
STUDIO オフサイド ☎0722(86)8050

★REINFORCER

ザインの今度の新作はトップビュータイプの8方向スクロールシューティング。敵工場などのステージを、3種類の武器を携えて駆けまわる。3種類のシナリオにしたがって主人公も変化し、各ステージ終了後など、ストーリーの分岐点ではビジュアルシーンによるストーリー進行が用意されている。

X68000用 5"2HD版 価格未定
ザイン・ソフト ☎0794(31)7453

★Gツール

ザイン・ソフトのグラフィックツール。最高512×512ドットまでのスプライト、BG、グラフィックキャラクターのデータを作成することができる。アイコンの編集やマウス両ボタンの機能の設定もできるので、効率的な作業が行えるぞ。

X68000用 5"2HD版 価格未定
ザイン・ソフト ☎0794(31)7453

★実践ビリヤード

PC-9801版「実践ビリヤード」がパワーアップしてX68000に登場する。プレイできる競技は四つ玉・ナインボール・ローテーション、そして最近注目のスヌーカーの4種類。練習モードや予想軌跡

の表示もできるので、初心者にもとっつきやすい1本といえそう。

X68000用 5"2HD版 9,800円
バック・イン・ビデオ ☎03(5565)8732

★プロミストランド

ポピュラスのシナリオデータ集が発売になる。といっても、単なるマップ集ではないぞ。思考ルーチンはもちろん、キャラクターや舞台まで一新されている。宇宙基地、フランス革命、LEGO、西部劇、コンピュータメーカー(あぶねー)、それと日本版だけのオリジナル、「江戸時代編」など。また違った気分の対戦が楽しめそう。ポピュラスファン必携だ。

X68000用 5"2HD版 4,800円
イマジニア ☎03(343)8911

★C-TRACE ver.3

レイトレーシングソフト「C-TRACE」がバージョンアップして登場。ボクセル分割により計算時間を短縮し、メモリ制限をなくして多量のデータのマッピングを可能にした。さらに、キーフレームソフトの追加やツリー構造の採用によって、モデリング、アニメーション作業の効率化が図られている。

X68000用 5"2HD版 98,000円
キャスト ☎03(705)1065

DōGA・CGアニメーション講座

寺田の教育的指導

今月の作品は、本職の日本画家という東京都の森山知己さん制作の“走る「MASAYA」と「メリーゴーランド（仮称）」、それと私こと寺田の作ったプログラム「BOMB.X」とその応用です（本誌103ページ）。

まず走る「MASAYA」は、人間の走っていくさまをCG上で再現したもの。関節の動きなどがとても忠実に表されているので、勉強になる作品です。

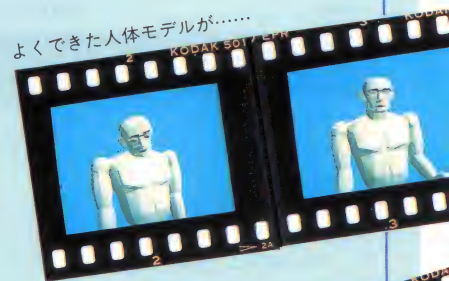
メリーゴーランドは、色づかいや雰囲気などメルヘンチックに表現されていて思わずほのぼのしてしまう作品です。さすがは本職の画家といったところでしょう。

「BOMB.X」は物体の爆発を題材にしたものです。応用作品も一緒に掲載しておきますので参考にしてください。

最後に「人体モデル」の画像データを示しておきます。これも人の動きを表したものです。リアルな感じが出ていると思いますのでじっくり観賞してください。



よくできた人体モデルが……



リアルに動くとさらに気持ち悪い……



●特集 マシン語への第一歩

ASSEMBLER PROGRAMMING

「マシン語」という響きは忘れられようとしている。「マシン語」は本来「アセンブリ言語」とは違う意味あいを持っていたはずだ。整った環境、高度なシステムサービス。68000では「マシン語＝暴走」という図式すら成り立たなくなった。それでもマシン語という響きが求められている。

マシン語をやっている人とそうでない人では歴然と違う。マシン語をやった人はとにかく強い。なんにでも強い。知識とか経験以前にとにかく強い。

8ビットではもうマシン語は必須科目といえる。68000の場合、これからの開発の主力は間違いなくC言語になるとしてもマシン語の知識は有用なものとなるはずなのだ。

42	ぜんまいちゃん再び	吉田幸一
47	マシン語ってなあに?	毛内俊行
52	MC68000の動作を探る	宮島 靖
CONTENTS 58	基本装備とおまじない	古村 聡
63	DOSコール&IOCSコールを使う	影山裕昭
67	マルチタスクへの挑戦	柴野雅彦
74	S-OS&REDAを忘れていませんか?	山田純二

ぜんまいちゃん再び

Yoshida Kouichi
吉田 幸一

吉田幸一氏の復活とともに、あのK君とぜんまいちゃんが帰ってきました。今回はぜんまいちゃんのマシン語に挑戦です。どうすればマシン語とつきあえるのか？ あれ、「ぜんまいちゃん」ってインタプリタの名前だけ……。

元来、コンピュータというものは素人さんが触れるようになるとは夢にも思わず、開発され使われてきたものである。しかし、時代は変わるもので、なんの間違いか素人さんがパソコンとやらを使うようになってきた。さあ、大変。いままではパソコンに触るのはコンピュータに興味のある人たち、大きなコンピュータは専門の技術者がいて、素人さんが触るのは技術者さんたちが作ったソフトやハード（銀行のATMとか）だったから、テキトーに作ってればよかったのだ。

そこで、素人さんが中を知らなくてもいいような新しいコンピュータが登場した。日本では“ファミコン”と“ポータブルワープロ”，アメリカでは“マッキントッシュ”だ。そんでもって、世のたいていのパソコンは素人さんが触るなんて考えてなかった

設計だったのだが、世界中で素人さんが使えるパソコンやソフトを！ てな感じで盛り上がりつつあったのであった。

ちょっと待った、と、誰もいわないから私がいおう。コンピュータって、そんな、着せ替え人形みたいにソフトを取っ換え引っ換えして便利になればいいものだったわけ。ゲームがたくさんあって、ソフトが多ければいいものだったけ？

違うのである。パソコンをただの着せ替え人形にしてしまったのは、コンピュータを特権階級のものにしておきたかったフォン・ノイマンの亡霊であり、国家の陰謀である。これからネットワークでいろんなものがデジタルでつながるようになると、その元であり端末であるコンピュータが重要なポイントになる。それをブラックボックスにしておきたい人がいるのだ。権力であり、企業である。彼らはコンピュータが一般市民にとってブラックボックスであることをいいことに、マインドコントロールをしようとしているのだ！ 立ち上がり！ ハッカーたち。都合の悪い情報を隠そうとする権力と戦うのだ！ はあ、はあ。

パソコンは個人が個人で個人を守る武器である。我々は権力に屈してはならないのだ。閉じた世界は銀行や商社のデカコンネットワークだけで十分。はあ、はあ。

パソコンの3つの顔

おっと、話が全然関係のない方向へ進んでいるぞ。元に戻そう。

パソコンはとにかくとんでもない代物である。そのパソコンには3つの顔がある。

ひとつは実務道具、事務機器としての顔だ。たとえばパソコンの能力を利用して、金儲けの手助けをしたり、面倒臭い虚礼的な手紙を書くのが面倒だからとひとつの文章を使い回したり、技術者が図面描いたりするときの顔だと思えばいい。まあ、たいていは従来あったものをよりよく管理しようというシミュレータ的な色彩が濃かつ

たりする。

もうひとつはいま流行っているメディアとしての顔だ。たとえば電腦倶楽部であり、ゲームであり、あなたが書いたグラフィックのデータであり、モデムを使って遠隔地とメッセージのやりとりをしたりするときの顔だ。ここで重要なのが、パソコンはテレビや本などの既成のメディアと違って情報の発信と受信の両方ができること、それからデジタルデータにできる情報なら絵だろうが文だろうが音だろうがなんでも扱えることだ。

さて、3つめの顔がコンピュータそのものとしての顔である。人間に対して挑戦的なほど論理的に動くコンピュータ。幾多の人間がその魅力に捕らえられたことだろう。パソコンはコンピュータの面白さに取り憑かれた人々の手によって発展したといってもいい。日本ではシャープが、道具やメディアとしてはイマイチだけど、コンピュータとしてはとても面白いパソコンを作り続けていた。ねえ、MZ君たち。

このコンピュータとしての顔を知らずして、パソコンの本当の面白さを知ることにはできないし、パソコンを値段分使いこなすことは難しい。世間ではメディアとしてのコンピュータが脚光を浴びているが、その陰でコンピュータそのものを楽しむ人がいなければならぬ生まれぬのだ。

もっとも重要なのはメディアとしてのパソコンとコンピュータそのものとしてのパソコンの2つだ。パソコンをコンピュータとしてしか楽しめないような視野の狭い人はいらぬよ、と釘をさしておこう。

と、いうわけで、私はコンピュータそのものを楽しむためのパソコン入門をここに展開することにしたのである。脆くてコピーしやすく盗みやすいデジタルデータを権力のものとしなないためにも、コンピュータを知る人たちを絶やしてはいけないのだ。

おっと、このままでいくとどンドン話が太く大仰になっていきそうなので、バキッと気持ちを入れ替えよう。



へろへろへろ。ふにやふにやふにや。はい、OK。

では、なんというか、何年も前にたった1回登場させたただけなのに、いまだに復活を望む声がちらほらする“ぜんまいちゃん”でもいってみようか。あーあ、なんか、書くほうとしては恥ずかしい。知らない人も多いだろうしなあ。別に知らなくてもいいんだけど。下手に書いてしまって、あーあ、きれいな思い出にしときたかった、てなことになるのもなあ。うーん。うーん。

決心。だいたい、きれいな思い出なんて食べるのがもったいなくて腐ったメロンのようなものである。そんなものに頼るやつがいけないのだ。

てなもので、ぜんまいちゃんとK君です。上の文章から察するに、K君がぜんまいちゃんを通じてコンピュータとはなにか、ど一すれば動くかという話へと発展していくのではないかと予想されます。

K君とぜんまいちゃんの話

月日のたつのは早いもので、K君はぜんまいちゃんユーザーになってもう3年になります。愛と希望のぜんまいちゃん。手垢にまみれたぜんまいちゃん。でも、女をくどくには金でも顔でも心でもなく、マメさが一番（第1法則）だと知っているK君は、ぜんまいちゃんにもマメです。手垢や汚れが気になり出すと、ぜんまいちゃんクリーナーで拭き拭きし、それでもだめならちゃんとメンテナンスセンターへ行って、皮膚の張り替えをします。もはやぜんまいちゃんはK君にとって、ただの馬鹿チンアンドロイドではないのです。ほら、猫嫌いの人でも3日も飼えば情が移るというではないですか。で、異性をくどくには情を移せば勝ち攻撃（第2法則）を図らずもぜんまいちゃんからくらってしまったのです。

そんなある日、K君はぜんまいちゃんのライバル機種「全自動“私を好きにして”ちゃん」（筆者注：おっと、これでは吉田戦車のパクリそのものではないか。吉田戦車さん、ごめんなさい）を買ったM君に大自慢されたのです。

「ぜんまいちゃんにお茶とお華ができるか。全自動“私を好きにして”ちゃんは、6000枚あるICカードでなんでもやってくれるんだぜ」

おっと、“私を好きにして”ちゃんは全自動というだけあって、ソフトさえ交換すればなんでもできるのです。

K君はといえばぜんまいちゃんをぜんま

いちゃんインタプリタという日本語ライクな言語で動かしていたので、複雑な動作だけでなく、高速な動作も苦手なものでした。しかも、微妙な動作を要求すると、フィードバックコマンドが速度についていかず、行ったり来たり“中風”状態になってしまふのです。

K君はハックおじさん（1987年5月号参照……するほどのこともないけど）を頼ることにしました。フラットなディスプレイに移ったおじさんの顔。K君は受話器に向かって、話の尾緒をつけて切々と語ります。

ハックおじさんはいます。

「それはぜんまいちゃんが悪いんじゃない。ぜんまいちゃんのCPUもハードウェアもお前がいったことができるくらいの性能はあるぞ。まあ、ぜんまいちゃんのハードを勉強でもして、直接動かしてみることもだな。機械を動かすには機械語が一番。おっと、愛人が来たので失礼するよ」

なんてアバウトな説明。

K君は「どーせ愛人だったってレプリカントだろうに」などと悪態をつきながら、うーん、とぜんまいちゃんを見つめます。確かに、ぜんまいちゃんが動くだけで感動した3年前とは違い、ただ日本語風言語で動くだけのぜんまいちゃんにはちょっともの足りなさを感じています。そのうえ、最近ではぜんまいちゃんがどうして“日本語の命令”で動くのか、疑問に思ってもいたのです。

ぜんまいちゃんの頭脳はモトール社のハイリスクプロセッサ、強欲1号です。超単純なコマンド体系といい加減な割り込み処理が特徴。

ぜんまいちゃんはデジタルなコンピュー

タですから0か1かの信号しか受け付けません。そのくらいはK君も知ってます。

「要は0か1か区別がつけばいいんだな」
とりあえず、あいさつをしてやろうと、JISコード表を調べ、「おはよう」を16進の「82A882CD82E682A4」にして、「1000001010100010000010111001101000001010100100」と2進数を書いてぜんまいちゃんに読ませました。もちろん、ぜんまいちゃんはそんなものを受け付けません。

1と0の並びを受け入れるのはあくまでもCPUの強欲1号であって、ぜんまいちゃんのCPUとK君のあいだには、CCDパターン認識カメラやらそれを解釈する信号プロセッサやらさらにそのデータを翻訳するソフトウェアなんかゴテゴテとあって、それではじめてぜんまいちゃんはなにかをするのです。この場合は、10000010101010001000001011001101000001010100100がそのままCPUに到達したのではなく、コマンド解釈・実行プログラムの入力チェックにひっかかってしまって、あえなく玉砕したのです。ま、当たり前ですけど。

「うん、そうくると思ったぜ」

K君はめげません。そんなとき、ハックおじさんから電話がかかってくるので、「どーせお前のことだから、指の1本も動かさなだらう」

まさにそのとおりで。

「でもおじさん、機械って1と0しか使えないんでしょ？ もしかして1011001100って、やんなきゃいけないの？」

「いや、それでもまだ考えが甘い。だって、機械にとってみれば10の1000001010100100という字と0という字があるだけじゃない

前回までのあらすじ

どじでのろまなK君は高校生の分際で“ぜんまいちゃん”という女性型アンドロイドを買ったのであった。うーん、金持ち。そんなもので、若者らしく“機械相手のエッチ”でもするのかと思いきや、なんと、ぜんまいちゃんに搭載されている言語でぜんまいちゃんを動かそうとするのである。おお、女と金にしか興味のないやつに読ませたいくらいだ。

で、「ぜんまいちゃんインタプリタVI.0」を使い、いろいろとぜんまいちゃんを動かすうち、見た目は人と変わらなくてもコンピュータはコンピュータであること、人間がいかに抽象的でもいいかげんで暗黙の了解と一般常識に頼って生きているかを知る（1987年4月号）のであった。

「ぜんまいちゃん」というのは富士山麓地下にある村正研究所で開発された女性型アンドロイド（全快1号）用の言語だが、全快1号自体の愛称として使われることも多い。ちなみに全

快1号の身長は160cm、美人というより可愛いというカンジの女の子仕様。フィリップ・K・ディック風というと、レプリカントかシミュラクラ……。得意技は「ぶーっ、間違っまーす」攻撃だ。

全快1号を開発した村正研究所の所長は自称“祝一平の永遠のライバル”祝一平である。「ぜんまいちゃんインタプリタVI.0」は正統研究所の開発による完全日本語仕様のインタプリタ言語だ。詳しくはOh!MZ1987年4月号を参照してほしい。もしかしたら、全快1号はもとより、元ネタになった「満開1号」「満開2号」を知らない人も多いのでは……という気もするので可能ならばOh!MZ1987年3月号も参照したほうがいいかもしれない。

* * *

当時私はMZ-2500のユーカラK2というワープロで原稿を書いていた。うーん、大昔のことのようだ。いまはX68000でHyperwordだもん。

か」

ハックおじさんはニタニタと“笑ウせえ
るすまん”笑いを見せます。

「じゃあ、どうすんの？」

「そりゃあ、だな、ぜんまいちゃんの背中
を開けると、スイッチがずらっと並んだパ
ネルがあるから、それをひとつずつオンに
したりオフにしたりするんだよ。そうすれ
ば、直接電気でぜんまいちゃんに伝わるぞ」

K君はア然としたまま石になりました。
おじさんはさらに、「で、だな、パチパチス
イッチを入れて、入力スイッチをペンツと
入れると、パチパチしたところを電気が流
れて、ぜんまいちゃんのメモリに流れるわ
けだな。で、作ったプログラムを流し込ん
だら、実行させればいいというわけだ」と
追い打ち。K君はそんな面倒なことをしな
ければならないとは露知らず、思わず「全
自動“私を好きにして”ちゃん」に買い換
えようかと思ったのでした。プログラミング
はできなくても、何千枚のICカードがあ
りますから。

でも、よく考えたK君。考えなくてもわ
かりますが、ぜんまいちゃんの背中にパネ
ルがあるなんてどのマニュアルにもネタ本
にも書いてありません。

「ところでおじさん、それ、本当？」

「わしが本当のことをいったことがあつ
たかね」

そして“笑ウせえるすまん”笑い。

「じゃあ、どーするんだよう」

「ちゃんと機械語でプログラム作る人の
ためのソフトウェアがあるから、安心しな
さい」

K君は「どーしてぜんまいちゃんが直接
理解できる機械語なのに、専用のソフトが

いるのだらう」と思いましたが、少しでも
楽できるのは（パネルのスイッチよりは）
いいわけで、「頂戴」といいました。おじさ
んは、電話回線（筆者注：この頃はデジタ
ル回線になってたりする。楽観的すぎるけ
ど）でそのソフトを転送してくれました。
そのソフトで作ったプログラムはぜんまい
ちゃんインタプリタとぜんまいちゃんの仲
立ちをしているぜんまいちゃんOSの上で
走ります。

「ついでに、スクリーン上で動作確認がで
きるぜんまいちゃんエミュレータもあるけ
ど、ほしいか？」

K君は、プログラムよりぜんまいちゃん
と遊ぶことが好きだったので、「いらない」
といいました。

ぜんまいちゃんを転ばせること

K君が一度やってみたかったことのひと
つに、ぜんまいちゃんを転ばせるというの
があります。ぜんまいちゃんには世界一優
秀なインテリジェントバルンサーが搭載さ
れており、それが働いてる限り、どんな姿
勢をさせてもバランスが崩れることはなく、
また、バランスが崩れるような姿勢は普通
に使っている限り無理なのでした。

K君が調べてみると、ソフトウェアでバ
ランサーのON/OFFは可能なようなので、
それをやってみようと思ったのです。

K君がまず調べたのはぜんまいちゃんの
CPU、強欲1号のコマンド。なんか、アセ
ンブラとかいうのを使うと0と1の並びを
打たなくてもいいそうで、なおかつ0と1
の並びでできることは全部やらせることが
できるそうで、なによりです。

サンプルプログラム

1:	エントリ:	
2:	ユーザー分岐	初期化
3:	ユーザー分岐	気をつけ
4:	移動	アドレス(\$ED001000), データ(\$FFFF)
5:	移動	アドレス(足上げのデータ), アドレスレジスタ1
6:	ループ: ユーザー分岐	右足, アドレスレジスタ1
7:	ユーザー分岐	バルンサー角度読み込み, データレジスタ1
8:	引き算	60, 値(データレジスタ1)
9:	条件分岐	式の値が零, ループ
10:		
11:	足上げのデータ:	* ボディコン用データセット
12:	データセット	\$08 * ボディコンデータ受信モード
13:	データセット	\$21 * 身体部分ID
14:	データセット	\$48 * 動作モード
15:	データセット	\$00 * 慣性モーメント
16:	データセット	\$CE * コントローラ1用ベクトル
17:	データセット	\$2A * コントローラ2用ベクトル
18:	データセット	\$99 * 対ショック対閃光防御
19:		:

で、強欲1号のコマンドですが、3種類
しかありません。演算と移動と分岐です。

「演算・移動・分岐。演算・移動・分岐」
K君は口の中で何度もつぶやいてみまし
たが、なにがなんなのか要領がつかめませ
ん。どうすれば「気をつけ」とか「お座り」
といった命令が“演算と移動と分岐”にな
るのか、そのあいだを埋めるものがサッパ
リなわけです。

ある日、『コンピュータの基礎』という
退屈そうな本を借りてきてやっと次のこ
とがわかりました。

演算というのは、論理演算と数値演算が
あって、普通は数値演算のことを計算、と
いうらしいこと。

移動というのはぜんまいちゃんが移動す
ることではなくて、メモリにあるデータや
アドレスを移動させたりコピーしたりする
こと。

分岐はメモリ上のいま実行している場所
を指すデータの内容を書き換えること。さら
に、条件分岐や割り込みによる分岐やユ
ーザー定義命令による分岐があるらしいこ
と。

K君の頭はクラクラ。初めてぜんまいち
ゃんインタプリタを動かしたときはマニ
ュアルを見ながら日本語の命令をいってや
ればなんとかなったのですが、機械語とな
ると、そうもいかないらしいのです。

そのうえ、これだけではどう考えてもぜ
んまいちゃんの指1本たりとも動かせる気
がしません。

「うーん、まだなんか秘密があるはずだ」

すつくと立ち上がってぜんまいちゃんに
向かって叫ぶK君。ぜんまいちゃんはスタ
ンバイモードでエネルギー消費を抑えてい
るので目を閉じたままうんともすんともい
いません。

ずるずるマニュアルを読んでもすと、
ROMというのがあって、それに“ぜんまい
ちゃんギアボックス”という名前がついて
いるのを発見しました。これは怪しい。
ROM。工場出荷時にはもうデータが詰まっ
ていて書き換えられないメモリというのが
怪しそうです。K君は「どーしてROMの反
対がRAMなんだろう。ROMがリード・オ
ンリー・メモリなら、もうひとつはRWM,
つまりリード&ライト・メモリのはずじゃ
ないか」などと思いましたが、これ以上悩
みを増やしてはいけないという理性の声に
よって留まったのでした。

で、ギアボックスと呼ばれるROMの中
にはどうやらぜんまいちゃんを動かすため
のプログラムが何千個も入っていたのでした。

でもK君は納得できません。だって、プログラムが何千個あろうが何万個あろうが、使っているのは演算と移動と分岐だけなはずです。

「これはまだ秘密があるに違いない」

そんなとき、ふと目に入ったぜんまいちゃんのカatalog。よく見ると、最新型ボディコントロールユニット搭載、とか、音声認識ユニットの高性能DSPとか書いてあるではないですか。そうです。CPUだけではないのです。

「うーん、問題なのは、CPUがどうやってボディコントロールユニット（略してボディコン）やら音声合成・認識ユニットやらに命令を出しているかだ」

パラパラパラパラ。K君、腕を伸ばして……。

「よう、こんな夜中にどうした。え？ まだあきらめてないのか。そーか、そーか。え？ そりゃあ、CPUがボディコンにボディコン用の命令やデータを渡しているだけだよ。ボディコンはそれを受け取って手足や腰なんかのコントローラに命令を送って、そいつらが実際に動かしてるわけ。どーすればできるか？ メモリの勉強をしたか？ アドレスがなにかわかってるか？ なんなら、教育用のコンピュータでも貸してやろうか」

「やだやだ、ぜんまいちゃんでなきゃやだ」

「んなら、簡単に喋ってやるから、あとでちゃんと勉強するように」

「へいへい」

「アドレスの一部がメモリの代わりにボディコンやら音声合成やらICカードコントローラやらとの入出力用に割り当てられ

吉田の謎

どーでもいい話だが、どーして私が約2年の間休んでいたのかというと、決して高橋留美子ファンに殺されるのが怖くて逃亡していたわけでもX68000を見捨てて98ユーザーになったわけでもない。某金融機関系ソフトハウスという固い職場でIBMちゃんの固いマシンの相手をしてたからである。

そして、大方の「そんなところ、お前に動まるわけがない。3カ月で辞めるほうに1万円!」という声にも負けず、めでたく2年ももったのであった。

当人は円満退職だったと思っているが、そうでない人もいたらしい。どーして辞めたのかというと、それは祝一平氏に「あやしい」といわしめたくらい謎である。ついでに、いま、なにを本職にしているかも謎である。

で、今後どーなるかも、ぜんまいちゃんがどーなるかも謎である。ま、いまの日本のアバウトで閉じた平和ほど謎なものはないけどね。



ていてだな、ボディコンだったら、ボディコン用のアドレスにボディコン用のコマンドやらデータやらを書いてやればいいわけだ」

K君はまたひとつお利口さんになった気がしました。これなら、演算と移動と分岐だけでぜんまいちゃんを転ばせられそう。直接ハードウェアを操作できるボディコン用の命令を、移動命令を使って、ボディコンが読めるところへ動かしてやればいいのですから。

ボディコンにバランスをOFFにする命令を送って、右足を少しずつ上げていけば、すぐ転ぶはずだ、と、K君はアルゴリズムを練ります。それなら簡単じゃん、と。

「よし、まずは初期化だな。最初に気をつけをして立たせておかないと駄目だからな」

で、気をつけはインタプリタにあるコマンドなので、絶対ギアボックスの中に用意してあると思って調べるK君。

ありましたありました。姿勢の初期化。「ユーザー分岐 気をつけ」。これをアセンブラにかけると「1010001110110011」というコードになって、ギアボックスの気をつけをさせるプログラムのあるところへ分岐して、そこでボディコンに気をつけをさせるための命令がダダダッと流れるのです。

そのあいだ、CPUは気をつけが終わるのを待って、終わったという信号を受け取ったら、

「気をつけをしたら、バランスのOFFだな。これはギアボックスにはないだろうから、直接ボディコンにコマンドを出さなきゃ。えっと、バランス制御のアドレスは16進でED001000だから、そのアドレスに、えーと、えーと、FFFFをセットしてやればよし、と。セットするのは移動コマン

ドだな。と」

「移動 アドレス (ED001000), データ (FFFF)」

「次は右足だな。「ユーザー分岐 右足アドレスレジスタ1」と」

これはアドレスレジスタ1に入っているアドレスの指しているメモリのデータをボディコンの右足制御コマンドのパラメータとして渡すといったものです。この場合のユーザー分岐のユーザーというのは、CPUメーカーにとってのユーザーということで、つまり、ぜんまいちゃんを作った村正研究所のこと。村正研究所が右足ときたら右足だぜ、ということにしたわけです。

「それには、右足を動かす前に、アドレスレジスタ1にアドレスをセットしなきゃならないし、その前に、そのアドレスに右足を動かすデータをセットしとかなきゃ、と。そんでもって、右足を少しずつ動かしたいから、ループさせて、と。ループ終了の条件も入れなきゃ。倒れても足を上げ続けるなんて醜いからな。壊れるかもしれないし。

「どーしようかな。えっと、バランスは姿勢制御をしないで、現在姿勢のデータは取っているはずだから、そのデータを読んで、60度くらいまで倒れたら実行を止めさせればいいだろう。それは、と、ギアボックスにあったから、大丈夫だな」

そんなこんなで、このあとぜんまいちゃんが暴走して100mを9秒で走っていったり、うんともすんともいわなくなったり鼻のリセットボタンを押したりしながら、なんとか図のようなプログラムができたのでした。

ぜんまいちゃんのOSは音声オペレーションですから、プログラムの名前をいってやれば実行してくれます。

「お転び!」

これがK君のつけたプログラム名。ぜん

まいちゃんは一瞬の内に右足を上げたかと思うと、ドタン、と、無事、K君を下敷きにして転んだのでした。K君、うまく動いて嬉しいやら痛いやら。

K君は腰のあたりを打ったせいか力ない声で「気をつけ!」と、コマンドをいいます。すると、ぜんまいちゃんはすつくと立ち上がります。

「はあはあ、この野郎、わざわざ俺の上に転ばなくても」と、ぜんまいちゃんに聞こえないような小さな声で囁いて立ち上がり、ぜんまいちゃんを「こいつう」とこづきました。

すると、軽く当たっただけにもかかわらず、ぜんまいちゃんはドタンと倒れたではありませんか。階下から「さっきからドタンドタンとうるさいわ!」と苦情もきてしまい、K君はあせります。

「そうか、バランスをONに戻し忘れていた」

そして、ガチャン。

「おーい、K。3日も学校へ来ないでなにやってんだあ」

と、ノックもせずに入ってくる悪友たち。代わる代わるぜんまいちゃんを撫で回します。

「お転び!」

ぜんまいちゃんはドタンとその悪友のひたりの上に倒れました。

「ぜんまいちゃんを転ばせるのに3日かかったんだ」

と、鼻の下を指でこすって、K君。

「こんなことで3日も休んでたのか、バカ

なやつ!」

そういつて彼の友達はずんまいちゃんの下敷きになってげらげらと笑っているのでした。

自分で作らなくとも

K君は謙虚にも人の作ったプログラムを読もう、と、決意します。転ばせるのに3日かかっていたら、ピアノを弾かせるには何年かかるかわかったものではありません。バランスをOFFにすればぜんまいちゃんはかなりの速度で動くことができるので、転ぶ心配のない上半身だけ使う技は超絶技巧が発揮できるのです。

そこで、ネットワークを覗いてみますと、ぜんまいちゃん用の機械語のプログラムがたくさんあるではないですか。K君は腕だけを使うプログラムを探しました。と、見ると、「シャープペン回し」というプログラムを発見。

作者のメッセージを見ますと、「親指と人差し指のあいだでシャープペンを回すプログラムです。何十年も前から受け継がれた技をぜんまいちゃんに引き継いでみました」となっています。K君はその技を知らなかったの(だいたいシャープペンシルなんてあまり使いません)、ダウンロードしてみました。

ダウンロードした中にはアセンブラのプログラム、実行プログラム、マニュアルがついてます。マニュアルによると、ぜんまいちゃんを座らせ、両手にシャープペンシルを持たせる必要があります。ちゃんと正しい握り方で。そして、回せ、という、ぜんまいちゃんが器用に両手で上手にシャープペンを回すではないですか。くるくる。くるくる。シャープペンシルが指の上で回転しているのです。

「おお、なんだ、この技は」

ぜんまいちゃんが微笑顔で(ちゃんと微笑むルーチンがプログラムに入っていた)くるくるシャープペンを回す様は異様です。

K君は自分でもやってみたくなったので、じっくりぜんまいちゃんの指を観察します。マニュアルには「速く」と「ゆっくり」と「やめれ」の3つのコマンドを受け付けることだったので、とりあえず、

「ゆっくり」

「ゆっくり」

2回くらいで、なんとか、指の動きがわかる速度になりました。

「ぜんまいちゃんって、凄い。こんなこと

もできるんだ」

できてどうするんだという話もありますが、K君はシャープペンシルを持って、ぜんまいちゃんの横で必死に練習するのでした。何度もシャープペンシルを落として、中の芯をぼろぼろにしながら。

やっとぜんまいちゃんと一緒に回せるようになった頃。

そして、ガチャン。

「おい、K、今日はなんで休んだんだ」

と、悪友はぜんまいちゃんを見ると、

「すごい。これ、お前がやらせたのか」

と、K君と一緒に回しているのを横目にいった。K君は、

「いや、ぜんまいちゃんがやってるのを見て俺が練習してるんだ」

「そんなことで休んだのか、やっぱ、バカなやつ」

そういうと悪友はけらけらと笑ったのでした。

K君はというと、おいおい、人のプログラムを見て勉強しようという意気込みはどうしたんだ。くるくる、くるくる。

* * *

と、いうわけであった。今回は高水準言語から機械語へ降りていくという従来見られたステップとは逆に、0と1からアセンブラへというもっと下から上っていく過程を書き加えたのだが、意図したところがわかってもらえただろうか。

機械語なんて言語だと思うからいけないのであって、あんなのはただの記号の羅列である。それをひとつずつ動作を考えながら積み重ねていく。これがはじめの一步。

アセンブラでバリバリプログラムを作れ、なんて私はいわない。ただ、一度でもアセンブラを経験し、コンピュータがどう動いているか、どういうことならできてどういうことはできないかを感覚的に知ることはすすめる。それは大変だけれども有用なことだ。どーせ、「ソフトはハードを壊せない」のだから、安心して暴走させればいい。そして、頭を使って悩めば少しは謙虚な気持ちになれるだろう。

さらに重要なのは、パソコンを使うのに必要なのは知識ではなく、知恵と感性だ、ということである。だから、ぜんまいちゃんのように完全架空の機械のほうが、知識に頼らないだけ、感覚を掴むにはいいかもしれない。

私はもう年である。ぜんまいちゃんもこれで終わりにしたいので、もうぜんまいちゃんがどーしたとかこーしたとかいわないように。では。

参考文献紹介

ラジカルなパソコン入門 岩谷宏著 ちくまライブラリー27 筑摩書房

この著者はマウスに偏見を持っている、PC-9801とIBM-PCしか認めていないような感がある(X68000のことなど知らないのだろう)、内容に多少怪しいところがある、パソコンを本文で書いた3つの顔のうちコンピュータそのものとしての顔としか見ていない(すべてその視点から語っている)といった気に入らないところはあるにしろ、内容はまっとうである。MS-DOS入門でもパソコンの歴史入門でもない、希有のパソコン入門書だ。ソフトウェアの著作権を認めない(それだけならただの馬鹿だが、この人は著作権自体認めないそうなので、それはそれで筋が通っている)という態度も面白い。著者の個性がはつきり出過ぎているので反発などもあろうが、初心者にはこびない良書だと見た。

戦え! 軍人くん 吉田戦車著 スコラ社

いうまでもないです。ちなみに、私は「友達の家へ泊まるっていつてきたの」ちゃんが好きです。

そのほか、P.K.ディックのさまざまな小説

マシン語ってなあに？

Mounai Toshiyuki

毛内 俊行

マシン語ってなんでしょう？ マシン語はコンピュータが唯一理解できる言語です。しかしマシン語は、我々のよく使うBASICやCとは、まったく異なった独特な仕組みや表現を持っており、初めて勉強しようとする人にとって、なかなか理解しにくい言語でもあります。

しかしほかの言語と比べると、プログラムの実行速度が桁違いに速く、サイズも小さくてすむというマシン語の魅力は、一度体験したらもう忘れられません。ここではそんなマシン語の概念を理解して、マシン語とは実際にどんな言語なのかを理解していこうと思います。

コンピュータの仕組み

現在のコンピュータは、ノイマン型コンピュータと呼ばれるもので、ドイツ系アメリカ人(だと思ったんだけど)のフォン・ノイマンという人が提唱した仕組みのものです。ノイマン型のコンピュータの特徴は、実行内容を記述したプログラムをメモリ上に置き、それをCPUが1つひとつ読みながら実行していくところにあります。

たとえば、メモリを大きなマンションだと仮定してみましょう(大胆!)。あなたがそのマンションへやってきました。最初に1号室の扉を開けるとそこに1通のメモが置いてありました。それには、

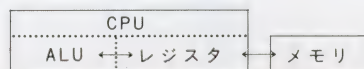
「3号室の扉を開けなさい」

と、書かれてあったので、あなたは3号室まで歩いて行き扉を開けました。するとそこにも1通のメモがありました。それを読むと、

「10という数を覚えておくように」

と、書いてあります。あなたは忘れないように自分の手帳に「10」と書き込んで、隣

図1 CPUとメモリの関係



の4号室の扉を開けました……。

さて、これはちょっと大胆な例ですが、コンピュータの中では、いつもこんなことが行われているのです。ここではあなたがCPUの代わりになって、メモリであるマンションの部屋を1つずつ訪ね、そこに置かれているメモを実行していきます。これがCPUとメモリの関係です。

メモリはマンションの部屋番号のように番地(アドレス)があり、各アドレスには00H~FFHの16進数が入ります。この数字1つひとつがCPUに対するメモとなり、このメモの羅列がマシン語のプログラムとなるわけです。

さて、それではCPUの内部をちょっと見てみましょう。CPUは図1に示すように、大きく分けてALUとレジスタ部に分かれます。ALUは演算装置のことで、ここであらゆる計算が行われます。そして、レジスタとは記憶装置のことです。

先ほどのマンションの例を思い出してください。たしか3号室へ行ったとき、10という数を忘れないように手帳にメモしたはずでした。レジスタはこのメモのような役割を持っています。そして、メモリとCPUはこのレジスタを介してデータのやりとりを行っているのです。参考までに、Z80のレジスタと68000のレジスタを図2に示しておきます。

仮想マシンBUG

今回は「Z80ユーザーも68000ユーザーも読めるマシン語入門」という、大胆不敵な原稿書きを申し渡されてしまいました。2つ返事で引き受けてしまったものの、どうしていいのかわからなかったのが、新しく仮想CPUを作ってしまうことにしました。その名もBUG(この名前に深い意味はありません)。これ以後、マシン語の概念的なところはこのBUGを使って説明していきます。BUGのアセンブリ言語はZ80のそれに似ています(これは私が68000のアセンブラ

世の中にはたくさんのCPUがあり、それとほぼ同じだけ違ったマシン語の作法があります。しかし、それらの根底となる部分はどれも同じようなもの。ここではZ80, 68000をまとめて、マシン語特有の考え方について解説します。

よりも、Z80のアセンブラのほうが使い慣れているためです)が、レジスタ構成は図3に示すように、ほかのCPUに比べてずっと単純です。

R0~R2は8ビットのデータレジスタ、A0は16ビットのアドレスレジスタだと考えてください。

そのほかに、フラグレジスタ(FLG)、スタックポインタ(SP)、プログラムカウンタ(PC)の基本的なレジスタが用意されています。さて、BUGの概要がわかったところで、早速マシン語の命令について説明していきましょう。

データの転送

まずはマシン語のもっとも基本的な命令で、なおかつもっとも頻繁に使われる命令である、データ転送命令について考えてみましょう。データ転送命令というのは、レジスタとメモリのあいだで、データの受け渡しをする命令です。たとえばBASICなどで考えれば、

HENSU=1

と、いった感じの命令です。BUGではデータ転送命令は、大きく分けて次の3つになります。

レジスタ ← 定数

レジスタ ↔ レジスタ

レジスタ ↔ メモリ

68000ではこのほかに、

メモリ → メモリ

といった内容の命令もあります。それではひとつずつ説明していきましょう。

・レジスタ ← 定数

BUGではデータ転送命令はLDを使います。たとえば、R0レジスタに1を代入する命令は、

LD R0,1

と書かれます。ただし、

LD 1,R0

と書くことはできません。これはBASICでR0=1は実行できるのに1=R0と書けな

いと同じです。転送命令では、転送先のレジスタが、必ず転送元よりも前に書かれてなければいけません(68000のアセンブラではこの位置関係が逆になります)。

さて、上の命令と同様に考えて、R2レジスタに50という数を転送する命令を考えてみましょう。そう、簡単ですね。

LD R2,50

と書けばいいのです。

・レジスタ→レジスタ

さて、次はレジスタ同士のデータの転送です。R0レジスタの中に格納されているデータをR1レジスタへ転送してみましょう。つまり、BASICでいうところのR1=R0を実行してやればいいわけです。これも転送命令なので、LD命令を使います。

LD R1,R0

これで、OKです。先ほどの定数を扱ったときと同じように、転送先、転送元の順でレジスタ名を書いてあげればいいのです。なおこのとき、転送といっても転送元の内容は消えてしまうわけではなく、転送後も転送元にデータが残ります。これはすべての転送命令において同じです。

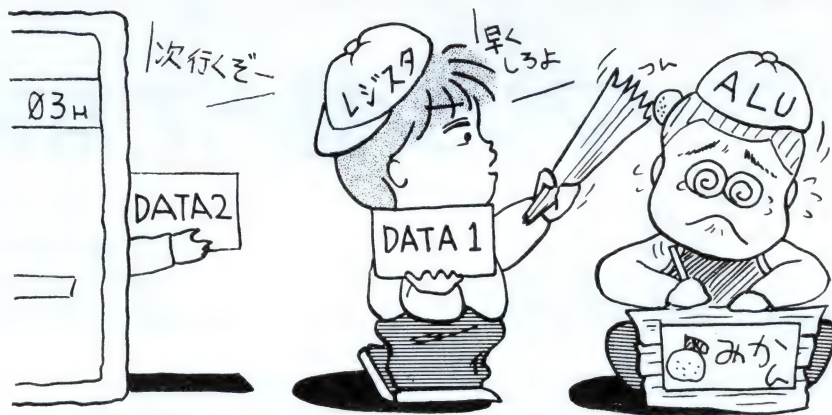
・レジスタ→メモリ

さて今度はレジスタとメモリのあいだでデータの転送を行います。今度は先に命令を見てもらいましょう。

LD R0,(\$F000)

さて、これは为什么呢？使われているのはLD命令ですから、転送命令に違いありません。するとR0が転送先、(\$F000)が転送元ということになります。これはR0レジスタに、メモリのアドレス\$F000に格納されている数を転送しなさいという命令なのです。このように数字を括弧でくくった場合、この数字で示したメモリ上のアドレスの操作を行うようになっていきます。

また、メモリアドレスの頭に付いている\$は、この数字が16進数であることを示しています。つまり、\$F001に格納されてい



るデータをR1レジスタに転送したい場合は、

LD R1,(\$F001)

と、なります。

また、メモリ・レジスタ間の転送命令は、逆にレジスタからメモリへデータを送ることもできます。つまり、

LD (\$8000),R0

と書けば、\$8000番地にR0レジスタの内容を転送するのです。

例題1 以下のプログラムを見て、次の問いに答えよ

LD R0,\$FF

LD R1,0

LD (\$8000),R0

LD (\$8001),R1

LD R1,(\$8000)

LD R0,(\$8001)

問 上記のプログラムを実行したあとでR0, R1レジスタに格納されている数を答えなさい。

解答

R0=0, R1=\$FF

※間違った人は間違いがわかるまでこの先を読んではいけません。

さて、まだデータ転送の話は続きます。BUGにはA0という16ビットのアドレスレジスタが存在します。今度はこれを使って

みましょう。

LD A0,\$F000

これは、A0レジスタに\$F000という定数値を代入する命令です。いままで使っていたR0~R2レジスタは8ビットのレジスタだったので、扱えるデータは\$00~\$FF(0~255)までだったのに対して、A0レジスタは16ビットなので、扱えるデータは\$0000~\$FFFF(0~65535)までとなりますが、それ以外に変わったところはありません。それではこれはどうでしょうか？

LD R0,(A0)

転送元が(A0)となっています。つまり、A0レジスタが示すメモリアドレスのデータをR0に転送するという命令です。

つまり、

LD A0,\$8000

LD R0,(A0)

というプログラムと、

LD R0,(\$8000)

というプログラムは、実は同じ動作をするプログラムになるのです。もちろん、

LD (A0),R0

と書いて、レジスタ→メモリと転送することも可能です。

演算命令

転送命令の次は演算命令です。BUGの演算命令はおおよそ次のようなものがあります。

加算(足し算)

減算(引き算)

図3 BUGのレジスタ構成

R0
R1
R2
FLG
A0
SP
PC

図2 Z80と68000のレジスタ

Z80のレジスタ構成

A	F	A'	F'
B	C	B'	C'
D	E	D'	E'
H	L	H'	L'
IX			
IY			
SP			
PC			
I	R		

68000のレジスタ構成

a0	d0
a1	d1
a2	d2
a3	d3
a4	d4
a5	d5
a6	d6
	d7
USP	
SSP	
PC	
SR	

それでは順番に説明していきましょう。

●加算命令

加算命令の代表が、ADD命令です。まず、下の例を見てください。

```
ADD R0,1
```

これは、R0レジスタに1を加える命令です。BASICで書けば、 $R0=R0+1$ となるのでしょう。基本的なところは先ほどの転送命令と同じで、LDの代わりにADDを使えばいいのです。つまり、

```
ADD R0,R1
```

は、 $R0=R0+R1$ となるのだし、

```
ADD R0,($8000)
```

と書けば、R0と\$8000番地のデータを足し合わせる命令になります。こうなれば当然、アドレスレジスタを使って、

```
ADD R0,(A0)
```

と書くこともできるでしょう。

もちろん、アドレスレジスタにも加算命令は使えます。

```
ADD A0,1
```

と書けば、 $A0=A0+1$ になります。

●減算命令

加算命令がADDなら減算命令はSUBです。使い方はADDと同じで、

```
SUB R0,1
```

で、 $R0=R0-1$ になりますし、

```
SUB R0,R1
```

とすれば、 $R0=R0-R1$ になります。同様に、

```
SUB R0,($8000)
```

で、R0から\$8000番地に格納されている数値を引き算しますし、

```
SUB R0,(A0)
```

で、R0からアドレスレジスタA0の示すメモリアドレスに格納されている数値を引き算します。もちろん、アドレスレジスタの減算も可能で、

```
SUB A0,1
```

で、 $A0=A0-1$ となります。

例題2 次のプログラムを実行したとき、以下の問いに答えよ

図4 BUGのフラグレジスタの中身



リスト1 ジャンプ命令とラベル

```
LD R0,1
LOOP:ADD R0,1
LD ($F000),R0
JP LOOP
```

```
LD R0,1
```

```
LD R1,5
```

```
LD A0,$8000
```

```
ADD R0,R1
```

```
SUB R1,2
```

```
ADD A0,1
```

問 レジスタR0, R1, A0に格納されている値を示せ

解答

R0=6, R1=3, A0=\$8001

ジャンプ&サブルーチンコール

さて、マシン語にだってGOTO命令やGOSUB命令のような、ジャンプやサブルーチンコールなどの命令があります。それぞれ説明していきましょう。

●ジャンプ命令

BUGでは、JPがジャンプ命令です。プログラム中で、

```
JP $8000
```

と書けば、コンピュータは実行アドレスを\$8000に移して、そこからプログラムを実行するわけです。まず、リスト1を見てください。リストの最後の行に、JP命令が使われています。しかし、JP命令の後ろには実行アドレスの代わりに、LOOPという文字が見えます。そう、ラベルです。X-BASIC以外のBASICではお馴染みですね。プログラムを見ると、2行目の先頭にちゃんと、LOOPという文字が見えます。このプログラムを実行すると、コンピュータがJP LOOPを実行したとき、実行はリストの2行目に戻るわけです。

●サブルーチンコール

サブルーチンコールはCALL命令を使います。使い方はジャンプ命令と同じで、

```
CALL $8000
```

と実行すれば、\$8000番地以降に書かれたプログラムをサブルーチンプログラムとして実行していきます。JP命令と違うところは、RET命令を実行したときに、実行が元のアドレスに復帰することです。これはちょうど、BASICのGOSUB~RETURNと同じです。

例題3 次のプログラムを実行したとき、以下の問いに答えよ

```
CALL SUB
```

```
JP JMP
```

```
SUB:LD R0,3
```

```
RET
```

```
JMP:ADD R0,4
```

問 R0に格納されている数を答えよ

解答 7

条件分岐

IF~THEN~のような条件分岐は、別に高級言語の特権ではありません。マシン語だってちゃんとできるのです。それにはまず、フラグレジスタの仕組みを理解してもらいましょう。

BUGのフラグレジスタは8ビットのレジスタになっていますが、使っているのは第0ビットだけです。図4を見てください。レジスタの第0ビットがZゼロフラグとなっています。ゼロフラグは、計算結果が0になったときに1になります。たとえば、

```
LD R0,1
```

```
SUB R0,1
```

と実行したとします。1行目でR0に1が入入されているので、SUB R0,1と実行したら、R0の中身は0になります。そのときにフラグレジスタのゼロフラグが1になるのです。

また、フラグは命令を実行するたびに刻々と変化していきます。そのため、

```
LD R0,1
```

```
SUB R0,1
```

```
ADD R0,1
```

と、実行したとすると、R0の内容は2行目で一度0になって、ゼロフラグが1になりますが、3行目で再びR0に1が加算され、R0の内容は0でなくなります。よって、3行目まで実行が終わった段階で、ゼロフラグは0となります。

Z80や68000などのCPUでは、このゼロフラグのほかにも、計算の繰り上がり、繰り下がりを示すキャリフラグや、オーバーフローの発生を知らせるオーバーフローフラグなど、たくさんのフラグが存在しますが、なにしろBUGは仮想CPUですから、もっとも単純なゼロフラグだけ用意しました。

さて、そのフラグの使い方ですが、条件付きのジャンプ命令というものが存在します。ゼロフラグに関するものは次の2つです。

```
JP Z,$8000
```

これは演算結果が0だったなら（ゼロフラグが1なら）\$8000へジャンプしなさいという命令です。そしてもうひとつが、

```
JP NZ,$8000
```

これは演算結果が0でないなら（ゼロフラグが0なら）\$8000へジャンプしなさいという命令です。この2つの命令で、条件分岐を行うことができます。

例題4 次のプログラムを実行して以下の問いに答えよ


```
LD R0,0
LD R1,10
LOOP:ADD R0,2
SUB R1,1
JP NZ,LOOP
```

問 R0, R1の値を答えよ

解答 R0=20, R1=0

その他のマシン語の動作

さて、いままではマシン語の概念を説明するうえで、BUGという仮想マシンを用意して話を進めてきました。途中、要所ごとに例題を用意しながら話を進めてきましたが、これらの例題が解けた人は、もうマシン語の概念は理解できたと思っています。

さて、ここらへんで一度BUGを使ったマシン語の解説は終わりにして、これからはもう少し違った角度からマシン語の説明をしていきましょう。

●スタックの動作

スタックという言葉聞いたことがあるでしょうか？スタックとはLIFO形式の構造を持ったデータ領域のことです。LIFOとはLast In First Outの略で、最後に入れたものを最初に取り出すデータ形式のことをいいます。

図5をご覧ください。まず、図のような箱を用意して、その中に球をひとつずつ落としていきます。図では最後に白い球を落とすことになっています。さて、これらの球は最初に箱に入ったものから順番に重なっているの、箱に全部球を入れたあとで箱から球を取り出そうとしたら、最後に入れた白い球を一番最初に取り出すことになります。つまり、最初に入れた二重丸の球を取り出すのは、一番最後になるのです。

このような構造をLIFO構造と呼びます。先ほども話したように、コンピュータのスタックは、このLIFO構造をとります。コンピュータのスタックは、SPレジスタ（スタックポインタレジスタ）の役割から理解しなくてはなりません。

SPレジスタはアドレスレジスタの一種で、その内容は常にメモリ上のあるアドレスを示しています。たとえばSPの内容が、\$F000だったとします。さてそこへ、コンピュータからスタックへ\$1234というデータが送られてきたとします。すると、アドレスを2引いた\$EFFFへ、いま転送されてきたデータが格納されるわけです。ただし、送られてきたデータは2バイトですから、スタックのメモリも2バイト必要です。



すると\$1234というデータの半分は、隣の\$EFFF番地へ格納されるのです。そう、スタックデータはだんだん若いほうのアドレスへと格納されていくのです（普通は）。

データをスタックにひとつ格納したとき、SPレジスタの値は\$EFFFに変わっています。ここで、\$8888というデータをスタックに格納すると、SPレジスタの値はさらに2つ減って\$EFCとになります。ここで、スタックからデータを取り出せば、最初に\$8888、その次に\$1234の順でデータが出てくるわけです。

さて、それではそのスタックはいったいどんなところで使われるのでしょうか？まず、さりげなく使われているのがサブルーチンコールをしたときのリターンアドレスの格納です。Z80でCALL命令を実行したとき、CPUは現在実行しているアドレス（の次）を調べ、そのアドレスデータをスタックに格納しているのです。そしてRET命令を実行したときに、スタックから取り出したアドレスへ実行を移しているのです。

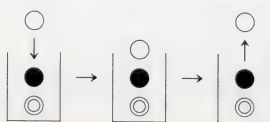
これは68000でも同じです。68000にはUSP、SSPと2つのスタックポインタが存在しますが、SSPはOSが、USPはユーザーが使うのが普通です。なお、68000ではCALLの代わりにbsr命令、RET命令の代わりにrts命令が使われていますが、名称が違うだけで動作にはなんの違いもありません。

さて、そのほかにもZ80には直接スタックを扱うPUSH、POPという命令があります。たとえば、

PUSH HL

と実行すれば、HLレジスタペアの内容がス

図5 LIFOの動作



タックに格納され、

POP HL

と実行すれば、スタックからHLレジスタペアにデータが転送されます。68000ではこのPUSH、POPに相当する命令はありませんが、データ転送命令（move命令）でスタックを操作することができます。

●論理演算の動作

さて、もうひとつコンピュータに欠かせないのがこの論理演算です。コンピュータで扱う論理演算には次の3種類があります。

論理積 (AND)

論理和 (OR)

排他的論理和 (XOR)

これらはそれぞれ、どのようなものなのでしょうか？まずはそれぞれの基本動作から見いきましょう。

基本的な論理回路は、図6のように2つの入力とひとつの出力回路を持っています。この2つの入力のパターンによって、出力が変わるのです。

●論理積

AND回路のことです。論理積はAとBの2つの入力が両方とも1のときだけ、出力Cを1にします。

●論理和

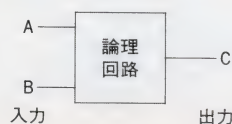
OR回路のことです。論理和はAとBの入力のどちらか一方でも1なら、出力Cを1にします。

●排他的論理和

いわゆるXOR回路のことです。排他的論理和は、AとBの入力の片方だけが1のとき、出力Cを1にします。

コンピュータでは扱うデータをよく、2

図6 論理回路の入力と出力



MC68000の動作を探る

Miyajima Yasushi 宮島 靖
Suzuki Yasuhiro 鈴木 康弘

マシン語とはCPUに対する直接的な命令です。CPUの動作を理解することがマシン語の理解であるともいえます。ここでは68000の基本的な命令とそれがどのように動いているのかを解説します。マシン語の動きをCPUの動作から見てみましょう。

CPUの世界

コンピュータの内部はとてつもなくデジタルな世界で、電圧が高いか低いかだけで動作しています。電圧が高いものを1、低いものを0とすると、0と1の2進数の世界になります。

CPUもデジタルなやつなので、0と1の組み合わせで動くように作られています。したがって、CPUになにか仕事をしてもらいたいときには、0と1の組み合わせで指示するしかありません。この0と1の組み合わせは、俗にいうマシン語のプログラムというものです。2進数8桁を16進数2桁で表すと、よく雑誌で見かけるダンプリストになります。

巷では、2進数8桁を8ビット、2進数16桁を16ビットなどと呼んだりします。16進数で2桁の数字は、8ビットです。8ビットをまとめて1バイトと呼んだりもします。

CPUは、このマシン語でしか動作しないので、BASICやCで書かれたプログラムもマシン語に変換されて（変換するのもマシン語のプログラム）実行されているのです。

メモリ

メモリとは、俗にいうRAM (Random Access Memory) やROM (Read Only Memory) のことです。メモリの中には、マシン語のプログラムや、いろいろなデータが入っています。メモリの最小単位は通常1バイト(=8ビット)です。そして、先頭から順番に番号がつけられています。一番先頭を0番地、その次を1番地、というように、番地という言葉を使って、場所を指定します。これはBASIC風に説明すると、メモリとは1つの要素が1バイトの大きさの配列変数みたいなものです。配列変数の添字が番地にあたります。68000ではこの番地に000000H~FFFFFFHまでを指定することができます。

しかし、CPUが基本的にできる作業は、整数の代入、四則演算、論理演算、条件判断、分岐くらいしかありません。

ですから、プログラムをマシン語で組もうと思うと、画面に文字を出すだけでも大変な作業になります。BASICのようにprint命令だけでなんでも表示できるようなお手軽さは、マシン語にはないわけです。

ですから、他人の作ったプログラムもちょっと見ただけではなにをしているのかわからず、1つひとつ追ってみたら文字を表示するプログラムだったなどということもあります。

アセンブリ言語

では、マシン語のプログラムを作るということは、CPUに与える2進数の組み合わせを作ることなののでしょうか。確かに、その2進数の組み合わせを作ってもプログラムはできてしましますが、2進数の羅列を直接入力していたのでは、画面に文字を表示するだけのプログラムでさえ挫折してしまうことでしょう。

では、実際にどのように開発を行うかというと、アセンブリ言語と呼ばれる言語でプログラムを書いていきます。アセンブリ言語がBASICやC言語などと根本的に違うのは、CPUが理解できる命令（先ほど説明したように、2進数の羅列ですよ）に1対1で対応しているというところ です。

レジスタ

CPUは、レジスタという、BASICでいう変数みたいなものを持っています。BASICの変数と異なる点は、BASICの変数はプログラマが自由に名前をつけられ、メモリの許す限りいくつでも作れ、浮動小数点

も自由に扱うことができるのですが、レジスタはCPUによって数と名前が決まっており、整数しか扱うことができないということです。

68000の持っているレジスタは、データレジスタ8本、アドレスレジスタ7本、プログラムカウンタ1本、スタックポインタ2本、ステータスレジスタ1本の19本です。ステータスレジスタが16ビットなのを除くと、ほかはすべて32ビットレジスタです。

データレジスタは、主に演算などに使用されるもので、BASICで使用する変数と、役割はほとんど同じです。D0,D1,……D6,D7の8本あります。

アドレスレジスタは、メモリをアクセスする際に、番地指定のために使われるレジスタです。A0,A1,……A5,A6の7本です。

プログラムカウンタは、次に実行される命令の入っている番地を指します。マシン語のプログラムはメモリに入っているわけですが、メモリにはほかのプログラムや、データなどもたくさん入っています。

CPUは、このプログラムカウンタが指す番地の内容を命令とみなし、プログラムを実行します。

また、命令を読み込むと、プログラムカウンタは次の命令の番地を指すようになっているので、CPUは常にプログラムカウンタが指す番地から命令を取ってくるだけよいのです。プログラムカウンタはPCと表記します。

スタックポインタとステータスレジスタはあとで説明します。

プログラムの終了

BASICでは、end命令を実行した時点でプログラムが終了します。また、endがなくともプログラムの最後（次に実行するも

のがない)まで行くと、自動的にプログラムが終了します。

しかし、マシン語ではきちんとプログラム終了の命令を書いておかないといけません。なぜなら、CPUはプログラムの最後がわからないからです。マシン語のプログラムはメモリ上に置かれ、先頭からPC(プログラムカウンタ)が順番に命令を指して実行されていきます。

プログラムの最後にきて、PCはその次の命令を指すようになっているので(プログラムの最後はプログラマがプログラムした最後ということであり、CPUはおかまなしに爆走している)、CPUはその次の命令を実行しようとしてしまいます。そこにどんな命令があるかは、プログラムを作った人にはわからないので、どんな動作をするかが見当がつかません。俗にいう暴走という状態になってしまうのです。

で、プログラム終了の命令なんですが、次の行をプログラムを終了させたいところに書いてください。

```
dc.w $FF00
```

これがなにを意味するかはあとで説明しますから、いまのところは気にせず、end命令なんだなということにしといてください。

なにか作ってみるぞ

では、とりあえずレジスタになにか数字を入れるプログラムを作ってみましょう。

```
move.l #0,d0
move.w #1,d1
move.l #2,a0
dc.w $FF00
```

これは、D0レジスタに0、D1レジスタに1、A0レジスタに2を代入するプログラムです。

“move”というのは、代入の命令です。

“l”というのは、扱うデータの幅を32ビットにする、という意味です。“#0”は数字の0を意味します。アセンブラでは、数字を表すときはこのように数字の前に“#”をつけます。“d0”は、レジスタのD0レジスタのことです。

“#0,d0”の部分オペランドといい、68000では左の“#0”をソースオペランド、右の“d0”をデスティネーションオペランドといいます。

“move”命令は、ソースオペランドをデスティネーションオペランドに転送するという命令なので、この場合0がD0レジスタに入ります。つまり、D0レジスタの内容を0にするのです。

BASIC風にと書くと、

```
d0=0
```

みたいな感じです。

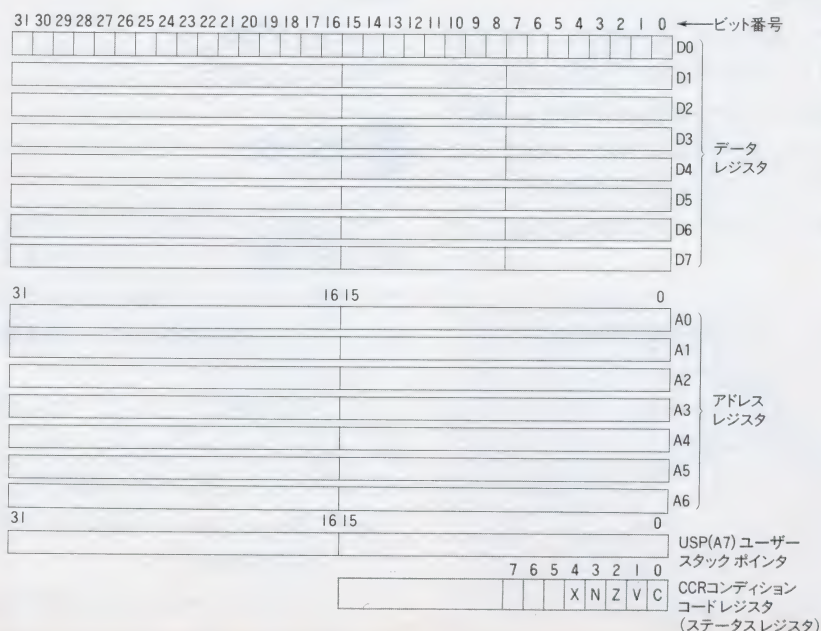
“l”の部分ですが、これはソースオペランドとデスティネーションオペランドのサイズを指定しているのです。“l”は32ビットのことです。68000は、数値を3つのサイズで表せるようになっています。ロングワード(32ビット)、ワード(16ビット)、バイト(8ビット)の3種類です。この場合、ロングワードで転送が行われ、D0レジスタの全32ビットが影響を受けます。もし、ワードの指定ならD0レジスタの下位16ビットのみ影響を受け、上位16ビットは影響を受けません。

ワードを指定するときは、“w”の代わりに“b”を書きます。また、バイトを指定するときは、“b”を書きます。

次の文は、1をD1レジスタに入れる命令です。ただし16ビットという指定があるので、結局D1レジスタの下位16ビットに1が入ります。D1レジスタの残りの上位16ビットは影響を受けず、この命令を実行する前の状態のままになります。

その次の文は、2をA0レジスタに入れる

図1



命令です。ちなみに、アドレスレジスタは必ずロングワード(32ビット)で実行されます。アドレスレジスタは、番地を指定するために存在するレジスタですので、ワードやバイトの操作は無意味というわけです。では、ワードで値を代入するとどうなるかというと、32ビットに符号拡張されてからロングワードで処理が行われます。

符号拡張というのは今回は説明を省きますが、要するに32ビットの値にするということです。この符号拡張の意味を理解しないで使わないほうがよいでしょう。32ビットに符号拡張されてから実行が行われるので、ワードでもロングワードでもCPU内部の処理速度は同じです。あと、バイトの操作はできません。ロングワードもしくはワードのみです。

さて、このプログラムを実行させてみましょう。このプログラムのファイル名をTEST.Sにしたのなら、

```
AS TEST.S
```

```
LK TEST.0
```

としてください。これで、ディスクにTEST.Xという実行ファイルが作られます。

実行は、

```
TEST
```

とするだけでOKです。

実行すると、すぐプロンプトが出てきますね。プログラムは実行されたのでしょうか？

このプログラムの場合、レジスタに値を入れるだけで、ほかにはなにもしません。したがって、レジスタに値を入れたあと、すぐプログラムを終了して Human に戻ってきてしまいます。

では、レジスタの値を表示させましょう。でも、マシン語には、BASICみたいな print 命令はないのです。そこで、DOSコールの登場です。さ、プログラマーズマニュアルを引っ張り出して、手元に置いておきましょう。

DOSコール

マシン語には、文字を表示する命令はないと書きましたが、文字を表示できないわけではありません。実際にいろんなところで文字を表示しています。

実は、X68000には、テキストVRAMというRAMがあり、ここに表示したい文字のデータを書くことによって、ハードウェアでその文字を画面に表示する仕組みになっているのです。したがって、なにか文字を表示させる場合は、このテキストVRAMにその文字のデータを書き込んでしまえばよいのです。この作業はプログラムにすると、結構面倒臭いものです。

テキストVRAMの構成は図2のようになっています。これは普通の16ビット機のグラフィックに相当する領域ですから、ここに適当なデータを書き込むと画面になんらかの模様を描くことができます。

まず、表示する位置を指定し、それを対応するVRAMのアドレスに変換します。次に表示したい最初の文字を取り出し、漢字コードから漢字ROM内のフォントデータ格納アドレスを計算して、最初のデータを先ほど求めたアドレスに書き込みます。さ

らに1ライン下のアドレスを求め、そこに次のデータを書き込んでいきます。これを16ライン分繰り返してやっと1文字表示の終わりです。

もし、文字に色がついていたら対応するプレーン分の処理を行い、強調指定があればデータをシフトさせたものと論理和をとって書き込んでいきます。さらに外字が使ったときのことも考えなければなりませんから、実際にはもっと面倒な処理が必要です。真っ先に外字にあたるコードかどうかを調べ、外字ならシステムの外字エリアを順に検索してフォントデータを得ることになります。

なにか文字を表示させるプログラムを作るたびに、この文字表示プログラムを作っていたのでは、夜が明けてしまいますね(プログラムは夜中に作ることが多い)。でも、実はあるんです。この文字表示プログラムが。

どこにあるのかというと、それはHumanの中です。ある方法で、このHumanの文字表示プログラムを簡単に呼び出すことができ、それを使うと、いとも簡単に文字を表示することができます。文字表示だけでなく、ディスクアクセス、キー入力など、いろいろなものが用意されています。これらを、DOSコールと呼びます。このDOSコールを自在に使えるようにならないければ、アセンブラでの開発は不可能です。ちなみに、先ほど説明した、プログラムの終了の命令の、

```
dc.w $FF00
```

ですが、これもDOSコールです。

スタック

さて、DOSコールを使えるようになるために必要な知識として、スタックがあります。このスタックは、普通にプログラムを組むときにも重要な概念なので、必ず理解しておいてください。

レジスタの説明のところで、スタックポインタなるレジスタが出てきましたが、このスタックポインタが今回の主役です。2つあると書きましたが、普通はそのうちのひとつしか使えません。

もうひとつは、HumanなどのOSが使っているのです。この、自分が自由に使える

スタックポインタを、SPまたはA7と書きます。アドレスレジスタはA0～A6で、スタックポインタはA7になりますが、このA7レジスタはほかのA0～A6とは性格異なりますので注意してください。以降、スタックポインタは、SPと表記します。

さて、スタックとはいったいなんでしょうか。スタックを説明するために、とりあえず以下のBASICのプログラムで説明します。

```
10 int mem(9999)
20 int sp=10000, x, y
30 push(10)
40 push(20)
50 x=pop( )
60 y=pop( )
70 print x-y
80 end
90 func push( x:int )
100   sp=sp-1
110   mem(sp)=x
120 endfunc
130 func pop( )
140   int x
150   x=mem(sp)
160   sp=sp+1
170   return(x)
180 endfunc
```

このプログラムを実行させると、10を表示します。なぜ10を表示するかわかると、もうスタックを半分理解したのと同じです。

関数pushは、spの値を1減らして、mem(sp)に引数を格納する関数です。

30行目のpush(10)の場合、spは最初10000ですから、sp=9999になり、mem(9999)=10となります。

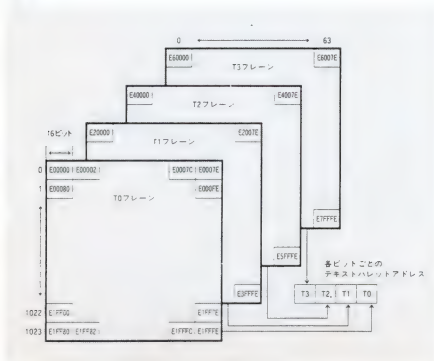
40行目のpush(20)の場合も同様に、sp=9998になって、mem(9998)=20となります。

関数popは、返り値にmem(sp)をセットして、spを1加える関数です。

50行目のx=pop()の場合、spの値は9998ですので、返り値がmem(9998)になります。これは40行目で代入した20です。その後、spに1が加えられ、sp=9999になります。したがって、xには20が入ります。

60行目のy=pop()の場合も同様に返り値は30行目で代入した10になり、sp=10000となります。したがって、yには10が入ります。ここまででx=20、y=10が入って

図2



いるので、70行目のx-yで10が表示されるのです。

関数pushを次々と呼ぶと、呼んだ順番のとおりそれぞれの引数が、配列変数memの最後の要素から入っていきます。関数pushの処理が終わったときのspの値は、最後にpushした添字になっています。

また、関数popを呼ぶと、pushによって入った値を次々と取り出していきます。関数popの処理が終わったときのspの値は、次にpopで読むところの添字になっています。

関数pushと関数popの処理が終わったときのspの値は、要するに有効なデータが入っている先頭を示しているということです。

なにを面倒臭いことを、と思っているでしょうが、これこそスタックなのです。配列変数memが、実際のメモリ（添字は番地になります）、変数spがスタックポインタになります。

要するに、スタックというのは、メモリのある番地に、データを代入していったり、そこからデータを取出していったりすることなのです。次々に値を代入していったり、スタックポインタがどんどん移動していきますので、pushによって以前にpushで代入した値が壊れることはありません。

この、pushの動作を、スタックに積む、といいます。先ほどのBASICのプログラムの30行目では、10をスタックに積んでいるのです。

このpushとpopの動作をするのは、アセンブラでは1命令になります。たとえば、10をスタックに積みたいときは、

```
move.l #10, -(sp)
```

と書きます。メモリはひとつが8ビットです。この場合、32ビットで行われるので、番地が4つ分必要です。したがって、SPは4減ります。また、レジスタもスタックに積むことができます。

```
move.l d0, -(sp)
```

で、D0レジスタの内容がスタックに積まれます。

スタックから値を取り出したいときは、

```
move.l (sp)+, d0
```

と書きます。この場合、スタックに入っていた値がD0レジスタに入ります。

さて問題です。次のプログラムの実行終了後、D0レジスタにはなにが入っているでしょうか？

```
move.l #1, d0
move.l d0, -(sp)
move.l #2, d0
move.l (sp)+, d0
```

答えは簡単ですね。1です。1行目で1をD0レジスタに入れ、2行目でD0レジスタをスタックに積んでいます。だから、スタックには1が積まれます。3行目で2をD0レジスタに入れているのですが、4行目でスタックから取り出した値をD0レジスタに入れています。したがって、1が入るわけです。

この場合3行目の文は無意味なのですが、ここで2行目と4行目の文によってD0レジスタの値が保存されることに注意してください。つまり、2行目より前と、4行目以降では、D0レジスタの内容は等しいのです。これは、よくレジスタの値が壊れてほしくないときに使うもので、3行目がD0レジスタを使う意味のある文で、その前後でD0レジスタの値が変わってしまつては困るときに使います。

ちなみに、D0レジスタだけでなく、D1もD2レジスタも壊れてしまつては困る、というときは、D0、D1、D2レジスタをスタックに積んで、あとでD0、D1、D2レジスタをスタックから取り出せばよいのですが、スタックは一番最後に積まれたものが一番最初に取り出されるということに注意してください。つまり、

```
move.l d0, -(sp)
move.l d1, -(sp)
move.l d2, -(sp)
```

でスタックに積んだら、取り出すときは、

```
move.l (sp)+, d2
move.l (sp)+, d1
move.l (sp)+, d0
```

となります。

さらにおいしいことに、実は複数のレジスタをスタックに積んだり、取り出したりするのに便利な命令があるのです。

たとえば、D0、D1、D2レジスタをスタックに積みたい、と思ったら、

```
movem.l d0/d1/d2, -(sp)
```

というように記述できるのです。また、こうして積まれたレジスタを復活させたいときは、

```
movem.l (sp)+, d0/d1/d2
```

と書けるのです。

D0、D1、D2 というように、レジスタの番号が続いたものを記述するときは、

```
movem.l d0-d2, -(sp)
movem.l (sp)+, d0-d2
```

でもOKです。

続・DOSコール

では、早速DOSコールを用いてなにか作ってみましょう。

DOSコールでスタックが必要なのは、DOSコールで引数を渡すのにスタックを用いているからなのです。たとえば、1文字表示のDOSコールに、表示させたい文字を渡すのには、表示させたい文字をスタックに積んでDOSコールを呼ばばいいのです。

1文字表示のDOSコールは、プログラマーズマニュアルを見ると、\$FF20の_PUTCHARです。コールのところを見ると、

```
move.w CODE, -(sp)
dc.w _PUTCHAR
addq.l #2, sp
```

と書いてあります。1行目は、表示したい文字コードをワード（16ビット）でスタックに積むことを表しています。2行目で、1文字表示のDOSコールを呼んでいます。これは、DOSCALL.MACを使わないならば、

```
dc.w $FF02
```

と書いても同じことです。

さて、3行目はなんでしょう。これは、SPに2を足すという命令です。アセンブラマニュアルに、“addq”命令の説明が載っていますので、見ておいてください。要するに、デスティネーションオペランド（sp）に、ソースオペランド（#2）を、ロングワードで足すということです。

これはなにをしているのかというと、1行目でワードで文字コードをpushしました。そのため、文字コードをpushする前に比べてスタックの値が2減っているのです。したがって、2を足しておかないと、スタックが文字コードのところを指したままになってしまうのです。

たとえば、次のプログラムを見てください。

```
move.w d0, -(sp)
move.w #$41, -(sp)
dc.w _PUTCHAR
move.w (sp)+, d0
```


これは、DOSコールの前後でD0レジスタをワードで保存しているつもりですが、DOSコールのあとのスタックの補正をしていないため、4行目では\$41がD0レジスタに入ってしまうことになります。

ちなみに、\$41ですが、これは16進数の41を表しています。16進数を表すときは、数字の前に\$をつけるのです。これはASCIIコードで'A'を表しています。

では、文字を表示するプログラムを作ってみましょう。とりあえず、'A'を表示させたいのならば、まず'A'の文字コードである\$41をスタックに積みます。そして、DOSコールをして、次にスタックに2を足して、スタックポインタの値を元に戻しておきます。

```
move.w    # $41, -(sp)
dc.w      _PUTCHAR
addq.l    # 2, sp
dc.w      _EXIT
```

プログラムの最後に、プログラム終了のDOSコールを入れるのを忘れないください。

このプログラムを実行すると、'A'が表示されます。この要領で、たとえば'X68000'を表示するプログラムを作ってみてください。

フラグ

さて、たとえば、D0レジスタが1だったら'1'を、1でなかったら'0'を表示させるプログラムを作りたいとなつたしましょう。1を表示したり0を表示したりするのは、先ほどのDOSコールで実現できますが、問題はD0が1だったら、というところにあります。なんとかしてD0が1かどうかを調べなければなりません。

そこで、レジスタのところで最後に出てきた、ステータスレジスタを利用することになります。このレジスタは別名フラグレジスタといい、フラグという1ビットのレジスタが集まってできたものです。フラグの中で、特に重要なのがゼロフラグと、キ

ャリフラグと呼ばれるものです。

ゼロフラグというのは、演算の結果が0になったときにセットされ(1になる)、0以外のときはリセットされます(0になる)。たとえば、

```
sub.l    d0, d0

```

という命令で(ちなみに"sub"というのは引き算をする命令で、デスティネーションオペランドーソースオペランドがデスティネーションオペランドに入ります)、この結果D0レジスタは0になり、ゼロフラグがセットされます。また、

```
sub.l    # 1, d0

```

という命令で(これはD0レジスタの内容から1を引く命令です)、1を引いた結果が0ならばゼロフラグはセットされ、0でなければリセットされます。

次にキャリフラグですが、これは演算の結果で桁上がり、または桁下がりが起こるとセットされ、それらが起こらなければリセットされます。わかりやすくいうと、キャリフラグは演算の結果が表せる数の範囲を越えるとセットされます。たとえば、バイトの演算では、表せる数は0~255です。したがって、加算をして255を越えたり、減算をして0を下回るとセットされるのです。ところで、もし255を上回ったり、0を下回るような計算をすると、レジスタにはなにが入るのでしょうか。たとえば、次のプログラムを見てください。

```
move.b    # $80, d0
move.b    # $80, d1
add.b     # $90, d0
sub.b     # $90, d1

```

これは、3行目で\$80+\$90、4行目で\$80-\$90を実行しています。両方ともバイトの指定がありますので、バイトで表せる範囲を越えています。したがって、両方ともキャリフラグはセットされます。これを実行したとき、D0、D1レジスタにはなにが入っているのでしょうか。

まず、D0レジスタ(\$80+\$90)を考えてみましょう。普通に演算すると答えは\$110です。こういうときは、下位8ビットの内

容がD0レジスタに入ります。つまり、\$10が入るのです。キャリフラグを第9ビット目と考えることができます。

では、D1レジスタ(\$80-\$90)はどうでしょうか。普通に演算すると、-\$10です。実は、-\$10というのは、\$F0と表せられるのです。なぜかという、8ビットの演算を考えた場合、\$F0に\$10を加えると、\$00になるからです(キャリフラグがセットされますが)。\$10加えて\$00になるのですから、-\$10と同じです。そういうわけで、D1レジスタには\$F0が入ります。

ちなみに、同様に-\$01は\$FF、-\$02は\$FEとなります。負の数をこのように表記することを、2の補数表示といいます。

次に、条件ジャンプ命令です。これは、ある条件が成り立つときのみジャンプするという命令です。条件というのは、たとえばゼロフラグが1だったら、とかゼロフラグが0だったら、というものになります。ゼロフラグが1ならジャンプする、という命令は、

```
beq      番地

```

ゼロフラグが0ならジャンプする、は、

```
bne      番地

```

キャリフラグが1ならジャンプする、は、

```
bcs      番地

```

キャリフラグが0ならジャンプする、は、

```
bcc      番地

```

という命令になります。

これを使うと、最初の、D0レジスタが1ならば1を表示させ、それ以外なら0を表示させるプログラムは、次のようになります。

```
sub.l    # 1, d0
beq      one
move.w    # $30, -(sp)
bra      other
one:
move.w    # $31, -(sp)
otehr:
dc.w      _PUTCHAR
addq.l    # 2, sp
dc.w      $FF00

```

1行目で、D0レジスタから1を引きます。もしD0に1が入っていたら、D0レジスタは0になり、ゼロフラグがセットされます。また、D0に1以外が入っていたら、D0レジスタは0にはならず、ゼロフラグがリセッ

図3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
T		S				I ₂	I ₁	I ₀				X	N	Z	V	C
																SR ステータスレジスタ

トされます。したがって、2行目で、D0に1が入っていたときは“one”に飛び、その他の場合はそのまま次の文が実行されます。

このプログラムの1行目で、D0レジスタが1かどうかを判断しているわけですが、このときD0の内容が書き換わってしまうことに注意してください。つまり、元々入っていた値よりも1少なくなっているのです。

引き算は、このようにある値を引くことによって、フラグを変化させることができます。しかし、レジスタに残る値は無意味です。この場合、D0レジスタの内容が1かどうかを調べただけで、D0レジスタに1少ない値が残る必要はないのです。どちらかといえば、D0の値はそのままのほうがいいでしょう。そこで、引き算をしてフラグは変化するけれど、引き算をした結果はレジスタに入らないという便利な命令があります。それは、“cmp”という命令です。たとえば、

```
cmp.l #10,d0
beq yahho
cmp.l #20,d0
beq haheho
```

のように使います。

このプログラムは、D0レジスタの値が10なら“yahho”に飛び、20なら“haheho”に飛ぶプログラムです。1行目でD0の値が変わらないので、3行目で20と比較するときもそのまま“#20”と書くことができます。これを“sub”命令を使って書くと、

```
sub.l #10,d0
beq yahho
sub.l #10,d0
beq haheho
```

となり、3行目で20と比較していることを理解するには1行目で10を引いていることを知っていないてはならないのです。

アドレッシングモード

いままで、“#0”と書くと0を、レジスタ名をそのまま書くとそのレジスタを表す、などと書いてきましたが、実はほかにもいろいろあります。この、“#0”で0を表す、こののをアドレッシングと呼びます。ここでは、基本的なものを紹介します。

まず、数値をそのまま書くとその番地の内容を表します。

```
move.b $10000,d0
```

この例では、\$10000番地の内容がD0レジスタに入ります。

次に、アドレスレジスタに括弧をつけると、そのアドレスレジスタが指す番地の内容を表します。

```
move.l #$20000,a0
move.w (a0),d0
```

これは、\$20000番地の内容をワードでD0レジスタに入れます。ワードですので、\$20000番地が上位8ビット、\$20001番地が下位8ビットになります。

これの発展したもので、括弧の前に‘-’をつけるとアドレスレジスタがサイズ分引かれてからその内容を表し、括弧の後ろに‘+’をつけるとその内容を表すのはその番地ですがそのあとサイズ分足されます。

```
move.l #$10000,a0
move.l #$20000,a1
move.w #100,-(a0)
move.w #200,(a1)+
```

この場合、3行目では\$10000からまずワード分の2が引かれ、A0レジスタに\$FFF Eが入ったところで\$FFFE番地に100が入ります。4行目では、まず\$20000番地に200

が入り、その後A1レジスタに2が足されます。

とりあえず、これだけ知っていれば大丈夫でしょう。ほかのアドレッシングモードはアセンブラに慣れてきてから自分で覚えていってください。

アセンブラプログラマへの道

ページ数の都合もあり、かなり急いで説明してきましたが、理解できたでしょうか。一度に全部理解しようとはせず、わかるところを少しずつ増やすような気持ちで勉強していきましょう。とにかくアセンブラをマスターするには、暇があったら、アセンブリ言語のプログラムを書いて、実行してみることです。そうすると、アセンブラ的な考え方がわかってきます。慣れてしまえばこっちのものです。

アセンブラでプログラムが書けるようになると、プログラムはなんでもアセンブラで書きたくなります。こうなれば、いやでもアセンブラの技術は上がっていきます。とにかく、毎日アセンブラに触っていることです。

ジャンプ

さて、X-BASICでは禁じ手となっている、goto 命令を知っているでしょうか。これは、

```
goto 行番号
というように使い、この命令があると、その行番号にジャンプするのです。
たとえば、
10 int a
20 a=10
30 goto 50
40 a=20
50 print a
60 end
```

というプログラムの場合、30行にくると、50行に飛んでしまい、40行を実行しません。したがって、50行で表示されるのは10です。

アセンブラにも、こういう命令があり、実行番地を指定して、その番地に制御を移すことができます。この命令は“jmp”です。たとえば、

```
jmp $10000
```

と書くと、この行を実行したら、次は\$10000番地から命令を実行していきます。

また、同じくある番地に飛ぶという“bra”という命令があります。これは、飛び先の番地が近いときにだけ飛ぶことができます。“jmp”命令より多少実行速度が速いので、とりあえず“bra”命令のほうを使います。

さて、普通アセンブラでプログラムを組むとき、どこ番地から実行されるなんてわかりません。したがって、“bra”命令でジャンプしたくても、ジャンプ先の番地はわからないのです。

そこで、アセンブラでは次のように書きます。

```
move.w #$41,d0
bra yahho
move.w #$42,d0
yahho:
move.w d0,-(sp)
dc.w _PUTCHAR
addq.l #2,sp
dc.w _EXIT
```

2行目では、番地を書く代わりに“yahho”と書いてあります。そして、4行目に“yahho:”というのがあります。これは、想像がつくと思いますが、2行目を実行すると、4行目にジャンプするのです。この“yahho”を、ラベルといいます。ラベルは、プログラム中の任意の行の先頭で、

ラベル:

と書くことによって、そのラベルを使っているところでアドレスに置き換わります（置き換えているのは、ASXです）。したがって、もし5行目の文が\$10000番地になったとしたら、2行目は

```
bra $10000
```

という文と同じになるのです。また、この場合で、

```
move.l #yahho,a0
```

というのがあれば、これは、

```
move.l #$10000,a0
```

と同じ意味になり、結果的に“yahho”の番地をA0レジスタに入れていることになります。

基本装備とおまじない

Komura Satoshi

古村 聡

ようこそ、マシン語の世界へ。だが、この世界での冒険へと旅立つ勇者よ、マシン語の世界もロールプレイングゲームの世界と同じようにそれなりの心構えとそれなりの装備が必要なのです。それにいろいろな手ごわい敵も潜んでいます。私はまだまだベテランとはいえないけれどそこそこ経験を積んでいるから役には立つと思いますよ。冒険へと向かうのなら私を仲間に入れませんか？

(で)を仲間に入れますか？(Y/N) Y

(で)は仲間になりました。

冒険の前にまず、装備を整えましょう。では、まずマシン語のプログラムを作るのに必要なものを挙げていきましょうか。内容についての解説がわからなくても名前だけは覚えてそれぞれ揃えるようにしてください。

では、まずはソフトから。

●ED.X

Human68k用のスクリーンエディタ。標準のシステムディスクに入っています。またC CompilerPRO-68K(以下XC), 福袋V2のパッケージにも含まれています。

●AS.X

68000用のアセンブラ。初代機であれば福袋のディレクトリに入っています。その他のACEやPROなどでは別売りで「福袋V2」「XC」に入っています。

●LK.X

同じくリンカ。アセンブラと同じく初代の「福袋」、別売りの「福袋V2」「XC」にあります。上級者はリンカとしてLK.XではなくPDSのHLK.X(ハイスピードリンカ)などを使っている人も多いようですが、標準のLK.Xがあればとりあえず十分。

●DB.X

同じくシンボリックデバッガ。「福袋V2」「XC」にあります。

●DOSCALL.MAC

DOSコールのマクロが入っているファイル。「XC」にあります。また「福袋V2」はついていません。またOh!Xの先月号のディスクのなかにも入っています。

●IOCSCALL.MAC

同じくIOCSコールのマクロ。「XC」、それとOh!Xの先月号のディスクのなかにも入っています。

マニュアル類としては、

●プログラマーズマニュアル

PROCESS.X, WHERE.Xなどのユーティリティ類の解説、DOSコール、IOCSコール、日本語FPのファンクションコールの呼び出し方、BASIC外部関数、デバイスドライバ、ユーザープログラムの起動後の状態などについての解説書です。X68000でプログラムを作る際にはこれがないと始まりません。「XC」「福袋V2」に付属します。

●アセンブラマニュアル

プログラマーズマニュアルと同じく「XC」「福袋V2」に付属するマニュアルでアセンブラ、リンカ、デバッガなどの取扱説明書です。またアセンブラの命令セットリファレンスも掲載されています。

基本的にはこれらのものが 필요합니다。よく見るとわかりますが必要なものはほとんど福袋V2かXCを買うとついてきます。つまりマシン語をやりたい人はとりあえず、

XCか福袋V2を買ってくる

が必要になるわけです。ま、どちらでも好きなほうを買えばいいでしょう。

マニュアルではないのですが、マシン語プログラムを組む際の参考資料として技術評論社の「68000プログラマーズハンドブック」や工学社の「X68000環境ハンドブック」などを買った人も多いようです。興味があるのなら、どういうものか見てみて気に入れば買ってみるといいでしょう。

冒険の書(その1)

お手ごろなところで「CRTのモードを512×512ドットにする」あたりからいってみましょうか(リスト1)。なに、初めてでも恐れることはありません。

まず、CRTモードを変えるためにはどうすればよいか、調べてみましょう。プログラマーズマニュアルが役に立つはず。調べてみるとDOSコール、IOCSコールそれぞれにCRTモードを変えるためのファンクションコールがあるのがわかりますね。DOSコールは、

conctrl (67~69pp.)

IOCSコールは、

CRTMOD (167~168pp.)

です。今回はDOSコールでいきましょう。

ところで、このリスト1を打ち込むわけですが、これはエディタで打ち込んだからといってそのまま実行できるわけではありません。冒険に出よう(プログラムを組もう)とする人がたいてい最初に使う言語B

リスト1

```

1:      *サンプルその1
2:      *      By Deono
3:      *      1990/05/22
4:      *.list
5:      *.include DOSCALL.MAC
6:      *.include IOCSCALL.MAC
7:      *.text
8:      _csize equ 16      * 画面サイズの変更モード
9:
10:     *プログラム領域
11: main: lea    usersp,sp      * スタックの設定
12:      move.w crtwork,-(sp)   * CRTモードを設定
13:      move.w _csize,-(sp)    * モードの設定
14:      DOS    CONCTRL        * CRTモード
15:      addq.l #4,sp
16:      DOS    EXIT          * 作業終了
17:
18:     *データ領域
19:      .data
20:      .even
21: crtwork: .dc.w 5           * ドットは512*512・65535色
22:      .ds.b 4096           * スタック
23: usersp:
24:      .end

```


ASIC。そのBASICのようなインタプリタと、ほかのほとんどの言語との最大の違いは、エディタがついてないということでしょう。BASICの場合には、BASICに入ったらそのまま、

```
10 int a,b /*変数の宣言*/  
というふうに、  
行番号 実行する内容+リターン  
とキーを押すことでプログラムを作って、  
RUN  
で実行していました。
```

ところが普通一般の言語というのはこの部分が一緒になってはいないわけです。たとえばアセンブラでコマンドライン用プログラムを組む場合、

- 1) エディタでプログラムを打ち込む
 - 2) アセンブラでマシン語のプログラムを作る
 - 3) コマンドラインでファイル名を打って実行
- というふうにステップが分かれているのです。

そこで、さっきの例にも出てきましたがプログラムを打つ部分を受け持つエディタというものが become になるのです。ここではエディタはED.Xを使うものとして説明をします。

まずはエディタの立ち上げ。エディタを立ち上げるのにはコマンドラインで、

```
A> ED  
とするかVSでエディタのアイコンをダブルクリックするわけですが、コマンドラインではそのとき一緒に打ち込むファイル名を指定できます。
```

```
A> ED CRT3.S  
このCRT3.Sというプログラム(このエディタで打ち込んだだけでまだ実行できないプログラムのことをソースファイルといいます)が存在していなかったら新規のファイルを、そうでなかったらその前に書いたプログラムの修正などができます。
```

EDで打ち込むときにはアセンブラのプログラムの場合は拡張子は必ず、

```
○○○.S  
のように.Sにしてください。
```

EDを使う

さて実際の打ち込みを行うわけですが、打ち込むだけならBASICとも感覚的にはそんなに違いません。次々とリストを(もちろん行番号はつけずに)打ち込んでリターン、打ち込んでリターンを繰り返すだけです。打ち込んだ行の修正は↑↓→←の

キーとROLL UP/DOWN・HOME キーでカーソルを移動して上書き(このときINSキーが押されていると挿入)したりDELキーで文字を削ったりすればいいわけです。

で、プログラムを組むわけですがリスト1には、

```
.list  
.include DOSCALL.MAC  
.include IOCSALL.MAC  
.text
```

というのがありますね。これはDOSコール、IOCSコールを使うファイルのリストが始まるよ……といういわばアセンブラに対するおまじないです。最初のうちはひとつ覚えてよいでしょう。ちなみに終わりのおまじないは、

```
.end
```

です。

最初と最後のおまじないのあいだにあるプログラムの中身はプログラマーズマニュアルにサンプルが出ていますのでそれを参考にしながら作っていきましょう。

プログラマーズマニュアルにはこのDOSコールのサンプルとしてこんなリストが出ています。

```
move.w y,-(sp) *MDにより異なる  
move.w x,-(sp) *MDにより異なる  
move.w #3,-(sp)  
dc.w _CONCTRL  
addq.l #6,sp
```

これを画面モードの変更に手直してやればいいわけです。

そして、プログラマーズマニュアルの69ページによれば画面モードを変えるには、
MD=16 conctrl(16,MOD)
MODで指定する画面モードにします。
MODの内容は次のとおりです。

```
：  
MOD=5 高解像度512×512,グラフィック65536色 (以下略)
```

だそうですから、さっきのサンプルの、

```
y,x を MOD に  
3 を 5 に
```

変えるだけでよさそうです。つまり、

```
move.w #5,-(sp) *mod  
move.w #16,-(sp) *  
dc.w _CONCTRL  
addq.l #4,sp
```

とすれば画面モードが変わるに違いありません。

気をつけてほしいのはMOVE命令で使っている値の頭に#がついていることです。68000のマシン語では変数の値(つまりメモリ上の値)を使うにはなにもつけないので

すが、直接2とか3とか数字を代入する場合には#マークをつけるのです。

実はサンプルの、

```
move.w y,-(sp) *MDにより異なる  
という行はYという変数の中身(Yが指すアドレスのメモリの持つ値)を代入していたのですね。
```

さて、ついでですからあとで読みやすくするために数字などのパラメータの部分を細工してしまいます。まず、マニュアルではDOSコールの呼び出し方を、

```
dc.w _CONCTRL  
としていますが、実はここはさっきのおまじないで、
```

```
DOS _CONCTRL  
と書くことができるのです。このほうがDOSコールだということがわかりやすくなるのでよいでしょう。
```

それから、BASICでたとえば、

```
conctrl(16,5)
```

を、

```
mod=5  
conctrl(16,mod)
```

と読みやすくするために代入したりすることがありました。これと同じことをMODを設定するのに使ってみましょう。マシン語ではプログラムと変数は分けておかなければなりません。このことに注意して考えてみるとさっきのプログラムなら、プログラム部分の、

```
move.w #5,-(sp)
```

を、

```
move.w mod,-(sp)
```

にして、データ部分に、

```
mod: .dc.w 5
```

というのを作ってやればいいわけです。データ部分が始まるおまじないは、

```
.data  
.even
```

で、プログラム領域の後ろにつけます。

それから代入に近いものとしてラベルに値を持たせることができます。つまり、たとえば_csizeというラベルを16という数字の代わりに使うことができるようにすることができます。そのためのおまじないがリスト中にも出てくるequ疑似命令です。

```
_csize equ 16 *
```

こんなふうに、

```
ラベル名 equ 値
```

というふうに書くことになっています。

注意が必要なのはこれは変数ではないためにこのラベルに数値の代入はできないということです。つまりさっきのmodに、

```
move.w #1,mod
```


(BASICでいうmod=1にあたる) はできませんが、

```
move.w    #1, _csize
```

ということ是不可能的です。

さて、あとはプログラムの実行を終わらせてコマンドモードに戻るだけです。プログラムリストの終わりを示す.endはリストの最後であることは伝えられるのですが、それはプログラムの終わりであるとはいっていないのです。プログラム自体はDOSコールで終わらせませす。

DOS _EXIT
がそれです。

さ、これでもいいプログラムは組めました。これからアセンブルに入ります。

アセンブルして実行だ!

次はアセンブラおよびリンカの使い方です。アセンブラやリンカの起動は、

A>AS ファイル名

A>LK ファイル名 ファイル名2……
です。

アセンブルの際にファイル名に拡張子をつけなければ勝手にアセンブラが.Sファイルであると解釈してくれます。したがって普通に作っているうちはファイル名に拡張子をつける必要はありません。

またAS.XとLK.Xにもスイッチがありますが、デフォルトでだいたいこと足りるのであまり使うことはありません。

AS.Xのスイッチ

/P……リスト出力を行う (デフォルトではファイル名.prnに出力)

/W……警告メッセージの出力禁止

LK.Xのスイッチ

/x……シンボルテーブルの出力禁止
まあ、こんなところでしょう。

では実際にリスト1をアセンブルしてみましょう。リスト1をCRT.Sという名前でエディットしているとしましょう。

このときエディタをいったん終わらせてもいいのですが、せっかくですからエディタからどうにかしてしまいましょう。例によっておまじないがあります。アセンブルのためのおまじないは、

ESC・H

ESC・C

AS CRT

ESC・C

LK CRT

これでアセンブルは終わりました。しかしなにかメッセージが出てきました。どうやらエラーがあってアセンブルできなかったようです。

X68k Assembler v1.01 Copyright 19
87 SHARP/Hudson

Warning: Line 13 absolute addressing

```
move.w    _csize, -(sp) *モードの設定
line 14 undefined symbol error
line 16 undefined symbol error
undefined symbol
CONCTRL
EXIT
```

2 Fatal error(s)

なるほど、14行目と16行目にエラーがあったようです。では14行目を見てみましょう。まず、カーソルを14行目にあわせませす。カーソルキーで運んでいてもいいですが、もっと簡単に探す方法がED.Xにはあります。ESCキーを押してから14、リターンという順に押してください。カーソルが14行目に飛んだでしょう。

さて、なにが間違えていたのでしょうか。そうです。ラベル名の前のアンダーバー(“_”)を忘れていたのです。IOCSCALL.MACやDOSCALL.MACではファンクションコール名の前に_が必要なのです。CONCTRLは_CONCTRL, EXITは_EXITとしなくてはなりません。

ほかにもなにやら警告が出ていますがエ

ラーではないということでもとりあえずほっておきましょう。ということで14, 16行を直したのがリスト2です。さ、もう一度アセンブルし直します。

もう一度ESC・Hからlk crt3 までのおまじないを打ち直しましょう。

またしても警告は出ていますが、とりあえずエラーは出ず、.Xファイルができたようです。一度エディタから抜けましょう。

ESC・E

これでエディタから抜けました。.Xファイルができていますのでコマンドモードでファイル名を打てば (あるいはVS モードでマウスをその.Xファイルのアイコンでダブルクリックすれば) 実行できます。

A>CRT3

なぜか、バスエラーが出てしまいました。やはり、先ほどの13行のワーニングがまずかったのでしょうか? 先ほどのワーニングの部分、13行をよく見てみましょう。なにかバグの原因になりそうなものがありますか? ありました。

```
move.w    _csize, -(sp) *モードの設定
```

_csizeはラベルに値を持たせたもので変数ではありません。直接、数値を書く (イミディエイトモードといいます。反対に変数 (メモリ上の値からとってくる) はエクステンドモードといいます) 場合は#マークをつけなくてはいけないのです。

このように必ずしもエラーメッセージがでないバグというのもマシン語のプログラムには多々あるのです。ではその部分を直したのがリスト3です。さあ、今度はさうまいくでしよう。エディタに入って書き直して、アセンブルして実行してみましよう。

A>CRT3

どうですか。今度はうまく512ドットモードの画面になったでしょう。やっとマシン語のプログラムが1本完成しました。あ

リスト2

```
1:      *サンプルその1
2:      *      By Decno
3:      *      1990/05/22
4:      .list
5:      .include DOSCALL.MAC
6:      .include IOCSCALL.MAC
7:      .text
8:      _csize      equ      16          * 画面サイズの変更モード
9:
10:     *プログラム領域
11:     main:      lea      usersp,sp      * スタックの設定
12:               move.w   crtwork,-(sp)  * CRTモードを設定
13:               move.w   _csize,-(sp)   * モードの設定
14:               DOS      _CONCTRL      * CRTモード
15:               DOS      _EXIT         * 作業終了
16:
17:     *データ領域
18:     .data
19:     .even
20:     crtwork:    .dc.w    5             *ドットは512*512・65535色
21:               .ds.b    4096          *スタック
22:     usersp:
23:     .end
```

リスト3

```
1:      *サンプルその1
2:      *      By Decno
3:      *      1990/05/22
4:      .list
5:      .include DOSCALL.MAC
6:      .include IOCSCALL.MAC
7:      .text
8:      _csize      equ      16          * 画面サイズの変更モード
9:
10:     *プログラム領域
11:     putank:      lea      usersp,sp      * スタックの設定
12:               move.w   crtwork,-(sp)  * CRTモードを設定
13:               move.w   #_csize,-(sp)  * モードの設定
14:               DOS      _CONCTRL      * CRTモード
15:               DOS      _EXIT         * 作業終了
16:
17:     *データ領域
18:     .data
19:     .even
20:     crtwork:    .dc.w    5             *ドットは512*512・65535色
21:               .ds.b    4096          *スタック
22:     usersp:
23:     .end
```


なたのレベルもひとつ上がったようです。

でも、512ドットモードになったはいいいが768ドットモードに戻れませんね。そうです、512ドットモードになるプログラムは作ったのですが768ドットモードに戻すプログラムを作っていないのです。

ついでにもう1本戻るプログラムを作ってみてください。なに、MODを5から0にするだけのことです。自分でやればもう一段レベルが上がることでしょう。

冒険の書(その2)

さて、もう1本プログラムを作って、いままで「おまじない」で済ませてしまったところを説明することにしましょう。リスト4はマウスカーソルを好きなパターンに書き直すプログラムです。このプログラムも例によってアセンブルしてX形式にして実行しなければなりません。基本的にはリスト3とまったく同じです。

```
.list
.include DOSCALL.MAC
.include IOCSALL.MAC
.text
で始まって、
.end
```

で終わっています。

まず、このあいだに作ったプログラムと違ってこれらは68000MPU自体の命令ではありません。これらは疑似命令といってアセンブラやリンカに対して行う命令なのです。アセンブラマニュアルでは213ページからの解説がそうです。

AS.Xには.dc命令のほかにもたくさんの疑似命令があります。そのなかでも一番よく見かけるのは、

```
.even
```

でしょうか。この命令は次の命令(moveや.dcなど)が偶数番地から始まるようにするものです。これはX68000で使用しているMC68000というMPUが「ワード以上の単位のメモリアクセスは偶数アドレスでしかできない」という制約によるものです。

68000MPUの命令はすべて2バイト以上の長さでかつ2バイト単位に大きくなるので奇数番地からはプログラムを動かすことができない。つまり、プログラムの先頭に必ずつけるようにしないと原因のわからないバグに悩まされることになりかねないからです。

.include命令はほかのマクロファイルを読む命令でサンプルリストでは、

DOSCALL.MAC

IOCSALL.MAC

をコールするのに使っています。これによってDOS&IOCSコールの呼び出し方を定義してあるマクロをリンカが呼び出しているのです。マクロの作り方がわかるようになったらぜひこのDOSCALL.MAC、IOCSALL.MACの中身を見てみるといいでしょう。

equは先ほども出てきたようにシンボルの値を決めるのに使います。この機能をうまく使うと、たとえば……、

extlop:

```
movem.l $e20000, -(sp)
movem.l $e20004, -(sp)
movem.l $e20008, -(sp)
movem.l $e2000c, -(sp)
movem.l $e20010, -(sp)
movem.l $e20014, -(sp)
movem.l $e20018, -(sp)
movem.l $e2001c, -(sp)
```

こんなリストを転送元の対象を\$E20000から\$E40000に変更したいときには、

extlop:

```
movem.l $e40000, -(sp)
movem.l $e40004, -(sp)
movem.l $e40008, -(sp)
movem.l $e4000c, -(sp)
movem.l $e40010, -(sp)
movem.l $e40014, -(sp)
movem.l $e40018, -(sp)
movem.l $e4001c, -(sp)
```

このように全部書き換えなくてはなりません。EQUを使って、

```
_txtvrm equ $e20000
```

extlop:

```
movem.l $_txtvrm, -(sp)
movem.l $_txtvrm+4, -(sp)
movem.l $_txtvrm+8, -(sp)
movem.l $_txtvrm+12, -(sp)
movem.l $_txtvrm+16, -(sp)
movem.l $_txtvrm+20, -(sp)
movem.l $_txtvrm+24, -(sp)
movem.l $_txtvrm+28, -(sp)
```

と書いておけば、シンボルの値を変えて、

リスト4

```
===== 4 =====
1:      *サンプル その2
2:      *      By Decno
3:      *      1990/05/25
4:      .list
5:      .include DOSCALL.MAC
6:      .include IOCSALL.MAC
7:      .text
8:      _ms_vpat: equ 6      * 表示するマウスカーソルの番号
9:
10:     *プログラム領域
11: main: lea     usersp,sp    * スタックの設定
12:       move.l msno,d1      * マウスカーソルの指定
13:       lea     patl,a1      * マウスカーソルのパターン
14:
15:       IOCS     _MS_PATST
16:       move.l   #_ms_vpat,d1
17:       IOCS     _MS_SEL
18:       DOS      _EXIT      * 作業終了
19:
20:     *データ領域
21:       .data
22:       .even
23: msno:  .dc.l   6          * マウスカーソルの番号は1番
24: patl:
25:       .dc.w    0,0
26:       .dc.w    %1111111111111111
27:       .dc.w    %1111111111111111
28:       .dc.w    %1111000000000000
29:       .dc.w    %1111100000000000
30:       .dc.w    %1101110000000000
31:       .dc.w    %1100111000000000
32:       .dc.w    %1100011100000000
33:       .dc.w    %1100001110000000
34:       .dc.w    %1100000111000000
35:       .dc.w    %1100000011100000
36:       .dc.w    %1100000001110000
37:       .dc.w    %1100000000111000
38:       .dc.w    %1100000000011100
39:       .dc.w    %1100000000001111
40:       .dc.w    %1100000000000111
41:
42:       .dc.w    %1111111111111111
43:       .dc.w    %1100000000000000
44:       .dc.w    %1010000000000000
45:       .dc.w    %1001000000000000
46:       .dc.w    %1000100000000000
47:       .dc.w    %1000010000000000
48:       .dc.w    %1000001000000000
49:       .dc.w    %1000000100000000
50:       .dc.w    %1000000010000000
51:       .dc.w    %1000000001000000
52:       .dc.w    %1000000000100000
53:       .dc.w    %1000000000010000
54:       .dc.w    %1000000000001000
55:       .dc.w    %1000000000000100
56:       .dc.w    %1000000000000010
57:       .dc.w    %1000000000000001
58:
59: crtwork: .dc.l   5          * ドットは512*512・65535色
60:       .ds.b    4096        * スタック
61: usersp:
62:       .end
```


_txtvrm equ \$e40000
とすれば変更も簡単です。

サンプルリスト中にはほかにも、
.list
.text
.data

があります。これにライブラリを作る際に必要な、

global

も含めて、この4つはぜひアセンブラマニュアルでも調べてみてください（アセンブラマニュアルを引いてみるいい練習にもなるでしょう）。

ところでこのプログラムにはマウスのパターンがデータ領域に2進数の定数で定義されています。リストを見てみると0と1でなにやら絵が描かれているところがありますね、それがそうです。

しかし、これをそのままリストの上から順に0,1で書き込んでいくとなにやら間違えてしまいそうです。BASICならこんな場合どうしていましたか？ 普通はなにか1行作っておいて行番号の部分だけ変えてリターン、行番号の部分だけ変えてリターン……とやって行のコピーをしたんじゃないかと思います。

これと似たようなエディタの機能として行のカット&ペーストがあります。ESC・P（エスケープを押してからPを押す）でカット、ESC・Gでペーストです。ESC・nPでn行カット、ESC・nGでn回ペーストです。これを使えばこういうときに、

.dc.w %0000000000000000

という行をカット（ESC・P）して32行ペースト（ESC・32G）して、その上に1で絵を書き込んでいく（もちろん上書きモードで）

ようにすれば簡単に展開できます。

いま、ESCキーを押して使う機能が出てきましたがED・XではよくESCキーを使用します。子プロセスの起動はESC・C（BASICでいう！コマンドです。ESC・Cとキーを押すとパラメータを聞きにきます）、

ESC・Hでファイルをセーブ

ESC・Yでファイルのマージ

ESC・Tでファイル名変更

ESC・Eでセーブして終了

ESC・Qでセーブしないで終了

などがあります。これで先ほどの「アセンブラのおまじない」のタネがわかったことでしょう。

データ長に注意

68000のアセンブラでプログラムを組む際に一番注意を要するのはなんといってもデータ長の問題でしょう。68000ではバイト（8ビット）、ワード（16ビット＝2バイト）、ロングワード（32ビット＝4バイト）という長さのデータを一度に扱うことができるわけですがそれぞれデータを扱う際にデータ長を指定しなくてはならないのです。

アセンブラでなんらかのテーブルなどを作るときにメモリ上にデータを置きたいなということがあります。それを実現するのがアセンブラ疑似命令のひとつ、

.dc

なのですが、このときもデータ長を指定しなければなりません。たとえば、ワードで100（16進数）という数字を置いておきたいときは、

.dc.w \$100

と書かなくてはなりません。

またバイト長データで10進数の100のときは、

.dc.b 100

とデータ長を.bにし、\$を取り去ります。

ちなみにロングワードで100（2進数）なら、

.dc.l %100

になります。

実は先ほどのマウскарソルのパターンも16進数で32個数字を並べてやっても構わないし、実際、単なる数値として扱われているのですが、2進数で描くとうまく絵のように見えるのでそのようになっているのです。

（で）の置き手紙……

さて、勇者よ。2つの冒険を経て君はもうひとりで立派にマシン語に挑むことのできる戦士として十分に経験を積んだことと思う。これからしばらく別々に行動しよう。ここから君が冒険に出るも、家に帰るも君しだい、というわけだ。しかしながら勇者よ、できればもっと成長した君とこのマシン語の世界でもう一度会いたいものだ。

BASICしかやってない君でもツールやマシン語プログラムのコツを一度覚えてしまおうと意外に簡単なもんだと感じたことだろう。それにひとつマシン語の世界を知っていればほかのマシンのマシン語の世界で冒険するのもそう難しいものではないし、またCなどに行こうと考えたときにも中身がなにをしているかがわかると非常にやりやすいだろうと思う。がんばってX68000でマシン語をマスターしてくれたまえ。幸運を祈っている……。

DB・Xは最後の武器だ！

マシン語のデバッグに欠かすことのできないものにデバッグ、db.xがあります。ではマシン語のデバッグではどんなことをするのか？ それはデバッグがどういう機能を持っているか知ることのできたいわかるのではないかと思います。db.xの主な機能は次のとおりです。

●1行アセンブル、逆アセンブル機能

●ブレイクポイント機能

●メモリ操作

●レジスタ操作

デバッグもコマンドラインで起動しますが、

A>DB ¥BASIC2¥BASIC.X

というふうにしっかりと、

ファイル名、拡張子

と指定しなければなりません。また当然のようにエディタからも、

[ESC]C

DB ファイル名

で起動できます。

とりあえず、

-G

でプログラムを実行します。

単なるバスエラーあたりだとこのコマンドで実行してみても変なアドレスをアクセスしてないか確認するだけでも結構効果があります。実際、リスト2をデバッグにかけてみると、

-g

Exceptional Abort By bus error

By Memory Access of 00000010

at 0008668C move.w \$00000010, -(A7)

と出てきて「確か16という数字は画面モードの変更をするときのパラメータだった→それがアドレスとして処理されている→イド抜けだ」というふうにすぐわかるのです。

あとデバッグの鉄則というのに「他人の目で見よ」というのがあって、早い話が自分で作ったプログラムを自分のものでないような気で見れば意外とバグが見つかりやすいのですが、その

とき役に立つのが、

-L アドレス

です。

これはアセンブラがソースから実行ファイルに変換したものを逆に実行ファイルをソースに戻す（逆アセンブル）ということをしてくれるのです。はたして、逆アセンブルなんか役に立つのだろうかというところ、これははっきりいってかなりの効果です。

ほかに68000でよくあるバグは次のとおり。even抜け……ワークエリア中にバイト長のデータがあってその後ろのワード長データをアクセスする際にアドレスエラーになることが多い。Xコマンドでレジスタをよく見るとよい。イド抜け……イミディエイトアドレッシング（#\$E28000）の際に#マークを忘れること。#マークが抜けてもエクステンドと解釈されるので見つけにくいことがある（特にアセンブラが警告を出さないようにした場合）。

DOSコール&IOCSコールを使う

Kageyama Hiroaki

影山 裕昭

今年も春になってX68000に新たに3機種の新製品が発表されました。高校、大学進学のどさくさにまぎれて親のスネをかじる人、あらかじめ予想して冬のうちからお金を貯めてた人、もうとくに新製品を購入してしまった人もたくさんいることでしょう。

なんにしても共通にいえことは、コンピュータを目的もなく買う人はいないということです。さて、あなたはX68000になにを求めているのでしょうか。ゲーム、レイトレーシング、MIDI、やろうと思えばなんだってできてしまうところが大きな魅力ですが、志半ばにしてすでに高価なゲームマシンとなりつつある人も少なからずいるに違いありません。

レイトレにしてもMIDIにしても市販されているソフトを利用することは時間的にも技術的にも大きなメリットがあります。しかし、自分でコンピュータを操作するという感覚に欠けることはいなめません。犬を飼うときも犬小屋を買ってくるのは簡単ですが、鋸とトンカチを使って自作して完成した犬小屋は、たとえ格好が不細工であっても、そのときの満足感はお金で味わえない貴重なものなのです。

ここでは皆さんにプログラムを作る醍醐味をわかってもらうためのアプローチとして、MC68000のアセンブラを使うときに最低限知っておいてほしいことをお話ししましょう。初心者の方にわかりやすく説明していくつもりなので、いままでアセンブラを使ったことのない人もぜひ最後までおつきあいください。

DOSコールとIOCSコール

マシン語はなんでもできる言語(?)です。ハードウェアを直接操作すれば、BASICやC言語で不満のある速度しか出なかった処理も高速実行可能です。しかし、このことを裏返せば、マシン語はなんにもかもしなければならぬ言語でもあります。

本来なら、画面にキーボードから打った文字を表示するだけでも、アセンブラの知識はもちろん、数多くのコントローラやサブCPUなどのハードウェアに通暁していなければなりません。

さらに「割り込みが発生したら」「エラーが起きたら」という不慮の事態に備えておかないととても安心してプログラムを実行できません。これもかなり面倒な処理です。

これではたまらないので、BIOSやOSというものが存在しているわけです。これらはシステムを管理すると同時に膨大なハードウェアを有効に使うためのさまざまなサービスを提供しています。

DOSコールとはHuman68kで使われているファイル入出力などの複雑な処理を、ユーザーが簡単に扱えるようにとOS側が提供しているものです。主な機能として、

キー入力
コンソール表示
プリンタ、RS-232C入出力
ファイル管理
かな漢字変換制御
ファイルの起動、プロセス管理

などがあります。

DOSコールはOSと密着しているもので、OSのリダイレクト機能を利用することも、もちろんできます。また、Human68kを使う限りプログラムの常駐、終了をするときもDOSコールを利用することになります。

DOSコールの利用には未定義命令であるラインFのエミュレータを使い、パラメータを必要とする場合はスタックを介してやりとりが行われます。見かけ上68000のマシン語にDOS管理用の命令が加わったような感じで使うことができます。

アセンブラでは次のように表されます。

dc.w \$FFxx

xxには00~FF_Hが入りますが、プログラマーズマニュアルを見てもわかるとおり、ところどころ数字が飛んでいます。これはDOSコールが将来拡張される時のこと

X68000でマシン語プログラミングを始めるには、まずX68000に用意されているDOSコールとIOCSコールを使ってみるのがよいでしょう。ここではサンプルを交えて各コールの使い方を中心に解説していきます。

を考えているのであって、実際にHuman v2.01では、初期のものに比べていくつかのDOSコールが追加されています。

他方、IOCSコールはX68000がROMで持っている基本入出力サブルーチンを利用するもので、X68000の持つすべての機能を使うためのサービスが用意されています。各種コントローラとのやりとりからマウス、グラフィック、割り込み制御までサポートされています。

DOSコールがOSと密着しているのに対して、IOCSコールはOSから独立しているのでどんな状態からでも使えます。

IOCSコールを使うには、レジスタD0にコール番号をセットして、MC68000のTRAP #15を使い、必要なパラメータの受け渡しには指定されたレジスタを使います。

```
moveq.l #$xx,d0
trap    #15
        (xxはIOCSコール番号)
```

DOSコールとIOCSコールがどういうものであるか少しはわかっていただけたでしょうか？

開発の手順

X68000によるアセンブラプログラムの開発は、

- 1) プログラムの作成、入力
- 2) アセンブル
- 3) リンク
- 4) 実行

の手順を経て行われます。実際には1回で仕様どおりの結果が得られることは稀なので、さらにデバッグという作業を経て、1)から繰り返すことになります。それでは最初はDOSコールを使ったプログラムの入力から実行まで、順を追って説明していくことにしましょう。

●入力

アセンブラで書かれたプログラムの入力にはエディタを使います。カレントドライブをBにして、

B>A:ED SAMPLE1.S

のように、これからSAMPLE1.SというファイルをBドライブに作成することを命令します。

早速リスト1を入力してみましょう。カーソルの移動や文字の削除などの基本的な編集操作は、BASICのプログラムを入力する場合とほとんど同じですから、難しいことはないはずです。

リスト中の*はプログラムに注釈をつけたいときなどに使うもので、*以降の文はアセンブルするときに無視されるものです。すべて入力し終わったら、ESCに続けてEと入力してください(以降ESC・Eのように表現する)。これでBドライブにSAMPLE1.Sが保存されました。

リスト1のようなアセンブリ言語で記述されたプログラムをソースリストと呼びます。ソース(source)とは「源の」という意味の英語からきているものですが、知らなかった人は覚えておいてくださいね。

●アセンブル

ソースリストを保存したら次の作業はアセンブルです。これにはアセンブラ(AS.X)を使います。アセンブラの役割はアセンブリ言語で記述されたソースリストを、CPUが直接わかるかたちである2進数(16進コード)に変換したり、メモリ領域を確保したりすることです。

B>A:AS SAMPLE1.S

としましょう。拡張子が.Sであれば、

B>A:AS SAMPLE1

でもかまいません。正常に終了すれば、

No Fatal errors(s)

と表示され、拡張子をOに変えたSAMPLE1.Oというファイルが新たにカレントドライブに作られます。

この拡張子がOのファイルをオブジェクトファイルと呼びます。先ほどカレントドライブをBに変更したのは、オブジェクトファイルをドライブ1に作成したかったからなのです。

さて、エラーがあった場合はエラーの発生した行番号と、エラーの種類が表示されますので、それをメモして再びエディタを起動して間違いを訂正して再アセンブルします。

●リンク

一般に数千行から数万行にも及ぶ大規模なプログラムを開発する場合などは、プログラムをその機能ごとにモジュールに分割して作成していくのがごく当たり前となっています。これらモジュールごとに分割された作成されたオブジェクトファイルをまと

めあげ、ひとつの実行可能ファイルに作りあげるのがリンカの役割です。

特にC言語では関数のほとんどがライブラリのかたちで提供されており、リンカはライブラリの中から必要なオブジェクトファイルを取り出すという重要な働きをしています。

ところが、このサンプルプログラムのように自己完結しているプログラムならリンカを通す必要はないと感じるかもしれませんが、そこはそれ。リンカを通さないと実行可能ファイルが作成されないことになっていますので、「アセンブルしたらリンカを通す」と覚えておいて間違いはありません。ではリンクしてみましょう。

B>A:LK SAMPLE1.O

アセンブルのときと同じように、拡張子がOであれば、拡張子を省略することができます。無事リンクが終了すると、拡張子がXになった、SAMPLE1.Xという実行可能ファイルがカレントドライブに作成されます。実行可能ファイル名を変えたいときはオプションを指定して、

B> A:LK /OA:TEST SAMPLE1
1のようにします。こうすればTEST.Xが

Aドライブに作られます。

●実行

実行は単純に、

B>SAMPLE1

とするだけです。このプログラムは画面に文字列を表示するものだったのです。まあ、ソースリストを見て容易に想像はついてい

たでしょうが。いままで説明してきたような手順をふんで、やっと実行可能なファイルを作成することができるのです。

プログラムの解説

さて実行できたところで、とりざたされていたプログラムの説明をすることにしましょう。

3, 4行目にあるのは疑似命令と呼ばれるものでMC68000にある命令ではなく、アセンブラが使い勝手をよくするために用意している命令です。dcやds命令などを除けば、疑似命令はオブジェクトコードに変換されることはありません。具体的には、.textはプログラムの始まりを指示するもので、evenはアドレスを偶数番地にローケ

リスト1

```
===== SAMPLE1.S =====
1: *      sample1
2:
3:      .text
4:      .even
5:
6:      move.l    #mes1,-(sp)    * 文字列の先頭アドレスを
7:                                *  スタックに積む
8:      dc.w      $ff09          *  _PRINT
9:      addq.l    #4,sp          *  スタックポインタを
10:                                *  補正する
11:      dc.w      $ff00          *  _EXIT
12:
13:      .data
14:
15: mes1:  dc.b      'SHARP X68000',13,10,0
16:
17:      .end
```

リストファイルの作成

大きなプログラムを開発していると、メモできないほどたくさんエラーが発生することがあります。そんなときはアセンブラのスイッチを指定すれば、リストファイル(アセンブラの出力するリスト)をディスクに作成することもできます。

B> A:AS /P SAMPLE1

とすれば、SAMPLE1.PRNというリストファイルがカレントドライブに作成されます。

リストファイルをうまく使うには、ED.XでSAMPLE1.Sを読み込んでおいて、ESC・Fと押し

ます。

編集ファイル:

と表示されましたね。ED.Xは同時に10個までのファイルを編集することができるので、これに対して、

編集ファイル: SAMPLE1.PRN

と入力します。すると、リストファイルが画面に表示されました。ここでエラーメッセージのある行を確認したら、ESC・A (SHIFT+F6) で編集テキストを切り替えて、ソースリストのエラーを訂正しましょう。訂正したら再び編集テキストを切り替えてリストファイルを表示して、ほかのエラー箇所を確認し、同様の作業をエラーがなくなるまで繰り返します。

似たようなものにタグジャンプというかなり便利なものがあるのですが、この機能は残念ながらAS.Xの出力するリストファイルに対しては使用することができないので、宝の持ちぐされといったところ

です。タグジャンプ機能を活用した効率的な作業には1990年6月号で村田氏が作成したようなアセンブラを使用することをおすすめしま

ションするものです。evenが必要な理由はMC68000が奇数番地にワードをアクセスしようとする、アドレスエラーを発生してしまうことに起因しています。この2つの疑似命令はソースリストで決まって使われるものなので、覚えておいてほしいものです。

6行目からMC68000のアセンブリ言語を使ったプログラムが始まります。このプログラムは文字列表示にDOSコール\$FF09の_PRINTを使っていますが(8行目)、最初にいったようにDOSコールでは引き数の受け渡しにスタックを使うことになっているので、引き数をスタックに積まなくてはなりません。この作業をしているのが6行目の、

```
move.l #mes1, -(sp)
```

です。#mes1の「#」はイミディエイトデータを指定することを表します。イミディエイト(immediate)とは「直接」という意味の英語で、単なる数値データを指します。

この場合はmes1がイミディエイトデータであることを表しています。mes1はプログラムが作成したラベルです。ラベルはプログラムでサブルーチンを作った場合や、このプログラムのように特定のデータを指し示す場合などにつけるもので、一般的には1カラム目から書き始め、最後にコロンのつけます(15行目)。

ラベル名をつける場合はそのサブルーチンやデータに関連性のある名前をつけるのがよいとされており、たとえばd0レジスタの値個だけスペースを表示するサブルーチンだったら、

```
prt_spc_d0:
```

のようにします。ラベル名にはプログラムのセンスが出てくるので、慣れていない人は本誌に掲載されているソースリストからラベル名のつけ方の「センス」を勉強すると思います。

続いてカンマの後ろの-(sp)は、スタックポインタの値を減じることを意味します。

いくつ減じるかは、move.lの?の部分によって決定され、b(バイト)なら1、w(ワード)なら2、l(ロングワード)なら4となります。逆に+(sp)なら増やすことになります。

結局、この命令が実行されると、スタックポインタを4減らし、そこにはmes1の置かれているアドレスが入ることになります。こうしてDOSコールを呼び出したあとでもスタックポインタは4減らされたままですから、スタックポインタに4を足してつじつまを合わせる作業が必要となります(9行目)。この一連のスタックの変化を図に表しておいたので参考にしてください。

11行目にあるDOSコールはプログラムの終了に必要なもので、サブルーチンの終わりをrtsで表すのに対して、プログラムの実行を終了するには必ずこのDOSコールを使います。

インクルードファイルを使う

次にリスト1と同じプログラムですが、DOSコールのファンクション番号をシンボルに定義したものがリスト2です。3、4行目のequがシンボルを定義するための疑似命令で、

```
ラベル: equ 式
```

で、以降ラベルの値が式の値と同値のシンボルとして扱うことが可能になります。

さらに変更を加えたのがリスト3です。リスト2でシンボル定義だった部分が、

```
.include doscall.mac
```

となっています。これはインクルードファイルとしてdoscall.macを使うことをいっているのですが、インクルードファイルってなんだ?と思う人もいるかもしれませんが。インクルードファイルとはアセンブラがソースファイルをアセンブルするときに挿入するファイルのことです。挿入するといってもアセンブルが終了すると除外されるものなので、ソースファイルの大きさが

アセンブルの前後で変わるようなことはありません。

ここで使っているdoscall.macはCコンパイラのシステムディスクのincludeというディレクトリに収められているもので、すべてのDOSコールのファンクション番号がリスト2と同じ要領でシンボル定義されているファイルなのです。だから、いちいちリスト2のようなシンボル定義しなくてもいいんですね。利用できるものはどんどん利用しましょう。同様に2つのDOSコールの表記方法が、

```
DOS <シンボル>
```

となっていますが、このDOSもdoscall.macで定義されているマクロ命令です。マクロとは1行から数行にわたる命令を1行で表せるようにと考えられたもので、doscall.macの中には、

```
DOS macro callname
dc.w callname
endm
```

とマクロ命令が定義されているのです。マクロはプログラムが見やすくなるという半面、サブルーチンと違ってアセンブルのときその都度展開されるものなので、あまり多用するといわずに実行ファイルを大きくしてしまうこともあり、注意が必要です。

インクルードファイルを使うときの注意点は、ソースファイルのあるディレクトリと同じディレクトリにインクルードしようとするファイルがないとエラーになってしまうので、違うディレクトリにある場合はアセンブルするときに、

```
B> A:AS /Iinclude SAMPLE3
```

と、スイッチでインクルードファイルのあるディレクトリを指定します。

このプログラムではさらにいくつかの変更点があります。まず、8行目がmove命令ではなくlea命令に置き換わっていますが、やっていることは同じです。一般的にアドレスをスタックに積むときはこちらの命令を使うことが多いようです。10行目もやっ

リスト2

```
===== SAMPLE2.S =====
1: * sample2
2:
3: _EXIT: equ $ff00
4: _PRINT: equ $ff09
5:
6: .text
7: .even
8:
9: move.l #mes1, -(sp)
10: dc.w _PRINT
11: addq.l #4, sp
12: dc.w _EXIT
13:
14: .data
15:
16: mes1: dc.b 'SHARP X68000', 13, 10, 0
17:
18: .end
```

リスト3

```
===== SAMPLE3.S =====
1: * sample3
2:
3: .include doscall.mac
4:
5: .text
6: .even
7:
8: pea.l mes1
9: DOS _PRINT *
10: lea.l 4(sp), sp
11: DOS _EXIT *
12:
13: .data
14:
15: mes1: dc.b 'SHARP X68000', 13, 10, 0
16:
17: .end
```


ていることは同じですが、addq命令では1から8の加算しかできないので、ほかのやり方も知っておいてほしいのであえて紹介しました。

* * *

ここでもう一度DOSコールを使う際の注意点をまとめておきましょう。

1) DOSコールが引数を必要とする場合はスタックに引数を積みます。

```
movem.l d0, -(sp)
pea.l LABEL
```

などが一般的な手法です。

2) DOSコールを呼び出します。

```
dc.w $FF09
```

のようにします。

3) スタックに積んだ分だけスタックポインタの値を増やします。

```
addq.l #4, sp
lea.l 10(sp), sp
```

IOCSコールを使う

いままで作ってきたプログラムをIOCSコールを使って書いてみたのがリスト4です。入力から実行までの手順はDOSコールの場合と同じですから、もう大丈夫ですよ

ね。見てもらえばわかるように、IOCSコールで引数が必要な場合はレジスタに値を入れることになっています。このプログラムでは文字列表示のためにIOCSコール\$21の—B—PRINTを使いました(9,10行目)。

引数には表示したい文字列が格納されている先頭アドレスをA1レジスタに入れることになっているので、8行目でそのようにセットしています。プログラムの終了にDOSコール—EXITを使うのに変わりはありません。

リスト6ではインクルードファイルとしてiocscall.macを使ってみたものですが、IOCSコールの呼び出し方が、

IOCS —B—PRINT

と、リスト4よりもすっきりしているのがわかるでしょう。これもiocscall.macの中に用意されているマクロ命令を使っているからこそこできることなのです。

いままで紹介したプログラムは単にDOSコールとIOCSコールの使い方を説明するために作ったプログラムなので、実行してもあまり面白みがありませんでした。そこで、最後に実用的かどうかは別にしても、IOCSコールだからこそこできるプログラムを紹介しましょう。リスト6を入力して

アセンブル、リンクしたあとに実行すると、あなたのX68000のROMの作成年月日とバージョンを表示してくれます。

使っているIOCSコールは4種類、ただひとつ使われているDOSコール—EXITはもうお馴染みですね。プログラムは力づくで作ったものなので、解析はおすすめてできませんが、IOCSコール—B—PRINTはエスケープシーケンスコード(45行あたりに見えている)も扱えるんだよ、ということとでバージョン表示をカラー文字にしてみました(趣味悪いかなー)。

今回使ったDOSコールやIOCSコールの個々の説明はCコンパイラなどに付属のプログラマーズマニュアルに詳しく書かれていますからそちらを参照してください。

これでひと通りDOSコールとIOCSコールの使い方はわかってもらえたと思いますが、文章だけではまだまだわからない部分もたくさんあると思うので、ぜひとも実際にサンプルプログラムを入力してください。ある程度アセンブラ言語に慣れてきたら、今度は自分でプログラムを作ってみる番です。初めからすんなりいくとは思いますが、己に負けず頑張ってください。期待しています。

リスト4

```
===== SAMPLE4.S =====
1: * sample4
2:
3: .include doscall.mac
4:
5: .text
6: .even
7:
8: lea.l mes1,a1
9: moveq.l #$21,d0 * B_PRINT
10: trap #15
11:
12: DOS _EXIT
13:
14: .data
15:
16: mes1: dc.b 'SHARP X68000',13,10,0
17:
18: .end
```

リスト5

```
===== SAMPLE5.S =====
1: * sample5
2:
3: .include doscall.mac
4: .include iocscall.mac
5:
6: .text
7: .even
8:
9: lea.l mes1,a1
10: IOCS _B_PRINT
11:
12: DOS _EXIT
13:
14: .data
15:
16: mes1: dc.b 'SHARP X68000',13,10,0
17:
18: .end
```

リスト6

```
===== ROM.S =====
1: * ROM.S
2:
3: .include doscall.mac
4: .include iocscall.mac
5:
6: ESC: equ $1b
7:
8: .text
9: .even
10:
11: IOCS _ROMVER * ROMバージョン
12: move.l d0,d1 * と作成年月日
13: move.l d0,d2 * を得る
14: andi.l #$00_ff_ff_ff,d1
15: *
16: IOCS _DATEBIN * BCD記述をバイナリ
17: move.l d0,d1 * 記述に変換
18: subi.l #$0050_0000,d1 * 西暦のオフセット
19: lea.l date,a1 * 80を引いて
20: IOCS _DATEASC * BCD記述を文字列に
21: lea.l date,a1 * 変換する
22: IOCS _B_PRINT * 作成年月日を表示
23: *
24: lea.l mes1,a1
25: IOCS _B_PRINT * 「バージョン」表示
```

```
26: swap d2
27: move.w d2,d3 * ROMバージョンの
28: lsr.w #4,d3 * 整数部を文字列にして
29: move.b #' ',d3 * ビリオドを
30: addi.w #$3000,d3 * くっつけて
31: move.w d3,ver * バッファに保存する
32:
33: andi.w #$0f00,d2 * 少数部を文字列にして
34: addi.w #$300d,d2 *
35: move.w d2,ver+2 * バッファに保存する
36: lea.l attrib,a1
37: IOCS _B_PRINT * ROMバージョンを表示
38: *
39: DOS _EXIT * プログラムの終了
40:
41: .data
42:
43: mes1: dc.b ' ROMバージョン ',0
44:
45: attrib: dc.b ESC,'[36m'
46: ver: dc.b 0,0,0,13,10
47: dc.b ESC,'[33m',0
48: date: ds.b 12
49:
50: .end
```


マルチタスクへの挑戦

Kuwano Masahiko

栗野 雅彦

Cだって万能ではない

低水準高級言語、あるいは高級アセンブラともいわれるC言語の普及により、アセンブラでゴリゴリとプログラムを作るのは処理速度を極めるような場合など、かなり特殊な場面だけになってしまいました。特にMC68000はミニコンのアーキテクチャを参考に作られたというだけあって、Cからアセンブラに比較的素直に変換されてしまうため、下手なプログラマがアセンブラで書くよりもCコンパイラでコンパイルしたほうがよほど綺麗で高速なコードが出るというようなことも珍しくありません。

システム資源のすべてがメモリ空間におかれる68000では、ポインタ変数さえ使えばどこでも直接アクセスできますから、アセンブラでできることはほとんどCでもできてしまうのです。しかも、制御構造の読みやすさや改造のやりやすさなどではアセンブラの比ではありません。

こうなってくると、いよいよアセンブラでプログラムを書く意義が薄れてしまいます。実際、私の場合も本当に処理速度上問題になるようなところやどうしてもアセンブラでなくては書けないような部分以外はすべてCになってしまっています。

このような状況でCでできるようなことをアセンブラで改めてやってみてもあまり面白くありません。今回はわずかに残されたアセンブラでしかできないことのひとつ、割り込み処理をとりあげ、簡単な時分割処理を行うプログラムを作ってみることにしました。

割り込み動作は個々のCPUに大きく依存するため、高級言語化がもっとも難しい部分でもあります。一部の8086用のCコンパイラやインテル純正(?)言語であるPL/Mなどでは、割り込み処理関数/サブルーチンの記述もできるようにしているようですが、やはりかなり無理があるようです。システムのほとんどがC言語で記述されたOSで

あるUNIXでも、割り込み処理の入り口などはアセンブラが使われていることから見ても、おそらくこの部分だけは当分のあいだ、アセンブラで記述され続けることでしょう。

割り込み処理

割り込み処理に入る前に割り込みの基本的な考え方について述べておきましょう。

CPUは基本的にメモリ上にある命令をひとつずつ読みとっては実行するだけの機能しかもっていません。いい換えれば、プログラムに書いてなければ、外部でなにかあると、一切応答しないわけです。コンピュータがいていうところの電卓、いわゆる「計算機」として用いられていたときにはこれでよかったのですが、より高度なシステムの一部として取り入れられるようになり、さまざまな周辺機器が増設されてインテリジェントコントローラとしても用いられるようになってくると、プログラム実行中でも外部からの信号にすばやく応答してもらいたい場面が出てきます。

このようなものに対応するためにプログラムのいたるところでこれらの信号をチェックするように作るのは不可能ではありませんが、プログラムを組むのは非常にやっかいになるうえ、ほとんどの場合は無用であるステータスのチェックに時間をとられるのですから、本来処理させておきたいプログラムの実行速度は低下、CPUの使用効率は著しく悪くなってしまいます。

X68000を例に取れば、キーボード入力やFM音源などがわかりやすいでしょうか。これらの要求速度というのは、CPUの実行速度からすれば、恐ろしく遅いものです。ワープロコンテストの入賞者で10分で千数百文字くらいというのを聞いたことがありますが、仮に文字として計算しても、1秒あたりたかだか3文字、入力文字数としても1秒あたり10文字にしかすぎません。

CPUのほうはといえば、2MHzのZ80で

少々高性能なコンパイラが現れても、どうしてもアセンブラを使わないとできないことというものもあります。ここでは、その代表例である割り込み処理を例にマシン語の醍醐味を味わってみましょう。題材は「I/OCSコールのマルチタスク実行」です。

も1秒に10万個程度の命令は処理できることを考えれば、いたるところでキー入力待ちをするというのがCPUにとっていかに無駄な作業かわかることでしょう。

このように非常にのんびりした要求でありながら、その応答速度はわりと厳しいのです。1秒に3回程度しか入力されないからといって、1秒に数回程度しかチェックしないと叩いてから一瞬待って、ようやく入力が受け付けられるという、耐え難いような応答になってしまうでしょう。FM音源なら時間がきたらすばやくレジスタの書き換えを行わないと妙な演奏になってしまいます。

このような要求に対応するために生み出されたのが、「割り込み」という考え方です。CPUに割り込み入力信号というものを付けておき、ここから割り込み要求を伝えると、CPUは現在行っている処理を一時中断し、あらかじめ用意された割り込み処理プログラムの実行を開始します。そして、処理が終わったところで特殊な命令(68000ならRTTE命令)を実行すると、再び先ほどまで実行していたプログラムの実行を継続するというからくりです。割り込みが入ったときに、割り込み処理からちゃんと帰ってこれるために必要な情報をセーブする作業はCPUが自動的に行いますので、プログラムを組む側で本来の仕事がどこまで実行されていたかを記録するようなことは必要ありません。

割り込みを使えば、外部からの要求がない限り本来のプログラムの実行を全力で行えますし、要求があれば現在実行している作業が終わりしだい、ただちに割り込み処理プログラムが実行され始めますので応答も非常によくなります。

図1にキー入力を例に割り込みを使わない場合と使った場合のプログラムの動きの違いを示してみました。割り込みを使わない場合には前回のキー入力チェックから次のキー入力チェックまでの期間はキー入力があったとしてもそれに気づくことができ

ませんから、応答をよくするためにはソフトウェアでこまめにキー入力チェックルーチン呼び出さなければなりません。これが(a)に相当します。

この例ではキー入力チェックにかかる時間1に対して、本来のプログラムが3の時間がたつたびにキー入力チェックを行っています。このため、CPUの処理速度は本来発揮できる速度の3/4になったのと同じことになってしまっています。

これに対し、キー入力を割り込みで行った場合が(b)です。この場合、キー入力がない限り、CPUは本来の処理を全速力で実行できます。しかも、キー入力があるからの応答は(a)の場合よりもよくなっているのです。

CPUの使用効率を下げずに外部からの要求にも的確に応答できるようになる割り込みという考え方はコンピュータにおける最大の発明ではないかと思います。

割り込みとマルチタスク

割り込みが入ると、それまで実行されていたプログラムの実行が強制的に中断されるということを積極的に使い、複数のプログラムを平行して動かすようにしたのが、時分割(タイムシェアリング)処理という手法です。割り込みを一定周期でかけるようなハードウェア(システムタイマとも呼ばれます)を用意して、この割り込みが入ったときにCPUが自動的にセーブした帰り先の情報などをこっそり別の場所に保存し、あらかじめ保存してあった情報と差し替えてなにごともなかったふりをして割り込み処理終了の命令を実行すれば、CPUはなににも知らず、別のプログラムの続きを実行するわけです。これにより、複数のプログラムがシステムタイマが入るたびに少しずつ実行されていくことになります。

この考え方を図2にしてみました。(a)のほうは、マルチタスクの考え方などとしてよく見かける図ですが、これではどうしてプログラムの実行が切り替えられるのかよくわからないので、もう少し正確に書いたのが(b)です。

割り込みが入ると、当然のように割り込み処理プログラムに移行するのですが、ここで帰り先を細工してやれば、RTE命令で戻るのはプログラムBのところになるわけです。CPUはそんなこととは露知らず、プログラムBのプログラムを実行していきます。そのうちまたタイマ割り込みが入り、今度はプログラムCを実行、次のタイマ

割り込みでは最初に保留されていたプログラムAに戻って、先ほどまでの続きを実行にかかるという具合になるわけです。

このような時分割の考え方をさらに一歩進め、並行して動くプログラム同士での情報交換や、あるプログラムからほかのプログラムの起動、ディスクやプリンタのように複数のプログラム同士のあいだで共有し

なくてはならない資源の管理などのサポートを行うようにしていったのが、マルチタスクOSと呼ばれるものです。時分割で実行されるプログラム1つひとつをタスク(task:仕事,作業)と呼んでおり、これらが同時に複数動くのでマルチタスクと呼ぶわけです。

パソコンの世界ではOSというとMS-DO

図1 割り込みを使った場合と使わない場合の違い

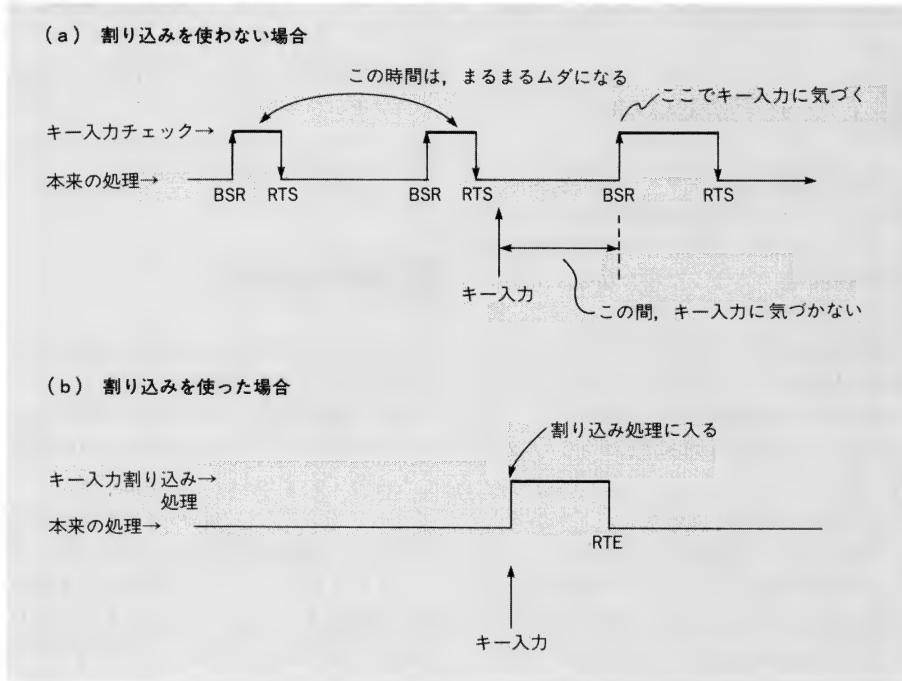
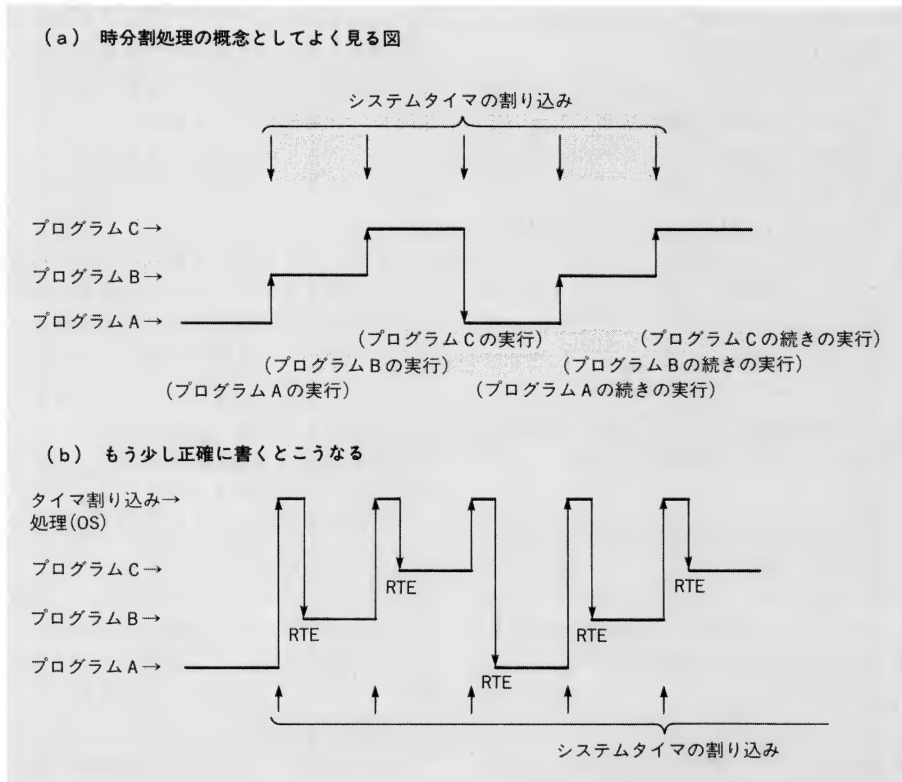


図2 時分割処理の考え方



Sが主流のようですが、これらは複数のタスクを動かす機能のない、シングルタスクのOSです。マルチタスクになるとシングルタスクのときには考えなくてよかったような、複数のプログラム同士の競合やシステムの持つメモリや入出力デバイスの取り扱いなどの管理に加え、タスク同士の通信などのサポートにも必要になるためOS自体の構造が複雑になるのですが、マルチタスク構造をうまく使ってやるとシングルタスクではやっかいだったことが大した苦勞もなく動かせたり、効率のよい動きをさせることができるようになります。

たとえば、シングルタスクのOSでファイルの中身をプリンタに打ち出すようにするとしましょう、プリンタに打ち出すプログラムは、ファイルを読み出してはプリンタに送りつけるというだけの単純なループですが、このあいだほかの作業はまったくできなくなります。これをいくらかでも回避するためにプリンタスプーラなどのソフトが作られてPDSでもずいぶん出回っていますが、独自のバッファを用意して、プリンタからの割り込みを横取りして……など、かなりの小細工をしており、単純にファイルを読み出して打ち出すプログラムとは似ても似つかないような内容です。

これがマルチタスクOSの下だったらどうでしょう。話は簡単です。ファイルを読み込んでプリンタに送るプログラムを新規のタスクとして起動してしまえばいいのです。

たとえばUNIXならコマンドの後ろに&記号をつけるだけでこうなります。コマンドを入力すると、新しいタスクとして打ち出すプログラムが登録されるだけで、すぐにコマンド待ちに復帰します。次のコマンドを入れればちゃんと動くところからも、プリンタへの打ち出しコマンドを実行している姿はまったく見えませんが、ちゃんとプリンタはジージーと鳴りながら動き続けているのです。

UNIXのコマンドなどを見ていると、さらにマルチタスク機能を便利に使っている例をいくらか見ることができます。たとえば、一般のパソコン用のCコンパイラは複数のパスに分かれていますが、これらをひとつずつ動かすよりないため、必ずディスク上に大きな作業用ファイルをいくつも作っては次のプログラムを読み込み、処理しては次のファイルを作り、前のを削除する、といったことを行います。このため、ディスクアクセスが非常に多くなることになります。

UNIXなどでは、Cコンパイラを起動す

ると、裏では必要なタスク（UNIXではプロセスと呼んでいます）を、ここではタスクで統一します）を一斉に起動し、互いのプログラムのあいだをパイプというタスク間の通信路で結んでしまいます。1つひとつのタスクは自分の入力として入ってきたデータを処理し、自分とパイプでつながれたタスクに渡していくという方法で、コンパイル作業が行われているのです。パイプで結んでおけば、複数のCコンパイラが同時に動かされたとしても（なにせ、マルチタスクですから）中間ファイルの名前がぶつかってわけのわからない動きをすることはありませんから、メモリをやたらと食うという以外はうまい方法といえるでしょう。

68000の割り込み機能

それでは、本命の68000の割り込みを見ていきましょう。68000の場合、これまで触れてきたような割り込み処理と同様の動きをするものとして、アドレスエラーやバスエラー（変なことをすると画面の真ん中にドローンと表示されることがありますね）、ソフトウェアのTRAP命令（X68000のIOCSコールはこれを使っています）なども含めた呼び方として「例外（エクセプション）」と呼んでいます。ここではハードウェアによる割り込みを中心に考えますので、「割り込み」という用語で統一していくことにします。

68000の割り込み動作はZ80や8086などよりもかなり凝った作りになっています。アーキテクチャ上、大きく違うのは、68000にはスーパーバイザモードと、ユーザーモードの2つの動作モードがあることと、割り込みに7段階のプライオリティ（優先順位）を設けていることです。

スーパーバイザモードではあらゆる命令

が実行できますが、ユーザーモードではシステムの動作に直接影響を与えるような命令の使用が禁止されます。さらにX68000では、ユーザーモードではVRAMを含め、各種の周辺コントローラやOSの領域へのアクセスが行えないようになっています。どちらの場合にも禁止されていることを行おうとした時点で自動的に割り込み動作が行われ、処理プログラムが実行されます。

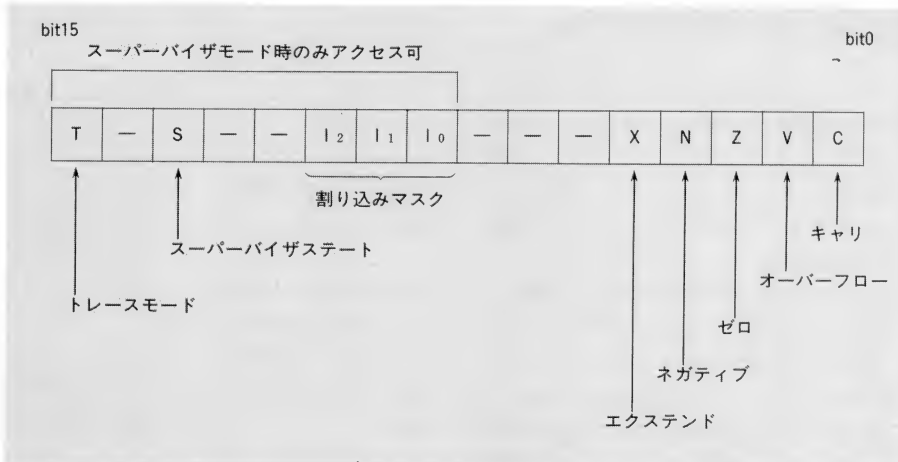
また、プライオリティ管理は、ステータスレジスタの上位バイトにある3ビットのマスクによって制御されます（図3参照）。Z80や8086では割り込みは禁止と許可の2段階しかありませんでしたが、68000では7段階まで拡張されているため、マスクビットも3ビット必要になっています。このビットは割り込みが入ったときにそのプライオリティと同じ値にセットされ、再び割り込み要求があっても、この3ビットで決められた以上のプライオリティでなければ受け付けられません。

逆にいえば割り込み処理中であっても、よりプライオリティの高い割り込みがあれば、それを受け付けるようになっているのです。これによって、ある程度ゆっくりと処理を行ってよい割り込みが処理されているために、緊急度の高い割り込み処理ができなくなるようなことを避けられることになります。

割り込みが入ってからの68000の動きはZ80のモード2割り込みや8086の割り込み動作とよく似ており、おおむね次のようになります。

まず、68000はただちにスーパーバイザモードに移行し、スタックポインタ（A7レジスタ）がスーパーバイザ用のもの（システムスタックポインタとも呼ばれます）とすりかわります。Z80や8086ではスタックポインタはひとつだけでしたが、68000ではモ

図3 68000のステータスレジスタ



ード別に別々のスタックポイントを持って
いるのです。

次に割り込み処理が終わったときに戻る
べきアドレスと、現在のステータスレジス
タの値をシステムスタックに待避します(図
4参照) 割り込み処理が終わったときに R
TE命令を実行すると、この番地に戻って
いくため、割り込まれた側のプログラムはな
にごともなかったかのように処理が継続で
きるわけです。

待避が終われば、次にCPUは割り込みベ
クタの読み出しを行います。ベクタ番号は、
今回の割り込みの要因がなにであるかを識
別するためのもので、値としては0からF
Fhのあいだになります。さらに、読み取
ったベクタをもとにベクタテーブルを参照し、
そこに書いてある割り込み処理プログラムの
アドレスを得ます。

ベクタテーブルはメモリの0番地から書
いてあり、アドレスはひとつあたり4バイ
トですから、(読み取ったベクタ)×4番地
の内容が割り込み処理プログラムのアドレ
スということになります。このあたりは80
86などとまったく同じです。

割り込み処理アドレスがわかれば、そこ
から順次プログラムを実行していき、最後
にRTE(ReTurn from Exception) 命令を
実行すると、システムスタックから帰る先
の番地と割り込みが起きた時点でのステータ
スレジスタの値を取り出してそこに戻るわ
けです。

68000で発生する割り込みとしてはこのほ
かにもTRAP命令など、ソフトウェアによ
るもの(DOSコールやIOCS コールを行う
のに使われています)、外部から異常が伝え
られるバスエラー、奇数番地からワードで
読み出そうとしたような場合に発生するア
ドレスエラーなど、数々の割り込み要因が
あります。

割り込み処理プログラム

割り込み処理プログラムに移ったとき、
CPUが自動的に待避してくれているのは、
わずかに帰る先の番地とステータスレジス
タの内容だけです。つまり、その他のレジ
スタ(A0~A6, D0~D7)はいままで動い
ていた状態のままになっているのです。も
し、これを割り込み処理の中で書き換えて
しまったらすると、どうなるのでしょうか。

割り込み処理から帰ったあと、実行され
るプログラムにとっては割り込みが入った
ことなどまったく知るところではありません。
プログラムが動いているあいだに勝手

にレジスタの内容が変わってしまい、その
まま実行を続けてしまうことになるのです
から問題です。運が悪ければ暴走にもつな
がりかねません。

むろん、割り込み処理プログラムの中で
まったくレジスタを使わないというのであ
れば、問題はないのですが、それではほと
んど満足な処理は行えないでしょう。した
がって、割り込み処理プログラムの中でま
ずやらなくてはならないことは自分の中で
使うレジスタをセーブすることです。

いちばん簡単なのは現在のスタック、す
なわちシステムスタックに全レジスタの内
容をセーブし、RTEを実行する直前に元
に戻してやる方法です。つまり、

```
int_entry:
    movem.l d0 -d7/a0-a6, -(sp)
    ;
    ;
    movem.l (sp)+, d0-d7/a0-a6
    rte
```

とやるわけです。これで戻ったときには全
レジスタの内容が変化せずにいることにな
りますから、無事になにごともなかったよ
うなフリができるわけです。雑誌に掲載さ
れるプログラムでも、バックグラウンドで
音楽を鳴らしたり、絵を動かすといった、
割り込みを横取りして動くようなタイプの
ものはこのような方法をとっています。

しかし、リストを見ていただければわか
るとおり、今回の割り込み処理ではこの方
法はとっていません。時分割処理を行うに
はこの方法ではうまくいかないのです。次
にこの点について検討してみることにしま
しょう。

時分割処理と割り込み

時分割処理を行うためには、2つのタス
クの状態を保存し、任意の側の前回まで
の情報を引き出し、その時点から再開で
きなくてはなりません。A、B、C、D4つの
タスクがあったとして、図4 割り込み発生時のシステムスタックの状態
Aが中断され、次にB
を実行するためには、
Bの前回までの状態を
保存しておかなくては
なりません。むろん、
次にBの実行が中断さ
れ、Aの実行をする
ためのAの状態も
保存しなくてはなりま
せん。

先ほどの例では、割

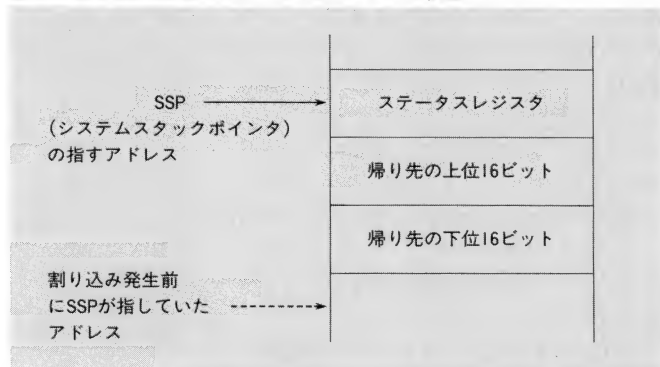
り込みが入った時点のレジスタの内容をす
べてシステムスタックに積み上げていまし
た。この方法を使つたとしましよう。いま
まで動いていたタスクをAとして、次にB
に移るためにはどうしたらよいのでしょ
うか。また、Bになんらかの方法で移った
として、いましがたシステムスタックに積
み上げたタスクAのレジスタの内容はどうす
ればよいのでしょうか。

Bのレジスタは、必ずいまのAのレジス
タ情報のさらに下にあるという方法を考え
れば、考えられなくもありませんが、いか
せん融通がきかない方法であることは間
違いありませんし、なにより貴重なシス
テムスタックが、各タスクのレジスタの待
避エリアに使われるのはなんと面白くあり
ません。

この解決策として考えられるのは、各タ
スクに1対1で対応するテーブルを作るこ
とです。選び出したタスクの実行を再開
させるときは、そのテーブルから情報を引き
出し、各レジスタの内容をすべて前回の時
点での値に復帰させたあと、RTE命令で戻
ればよいわけです。このようなテーブルを
TCB(タスクコントロールブロック)とか、
プロセステーブル (UNIXの場合)と呼ん
だりしています。

レジスタの情報の待避する場所としてよ
く使われるのはTCBや各タスクのスタッ
クです。スタックに保存した場合は、最後
のスタックの位置だけをTCBに保存してお
けば、残りのレジスタの値はそこから順に
引っ張り出してくることができるようにな
るわけです。どちらを使うかは設計者しだ
いといったところでしょう。

TCBにタスクのすべての情報を残すとい
う点ではレジスタの内容を全部保存するほ
うがよいでしょうが、すでに保存領域が指
し示されているのだからスタックを使うほ
うが簡単であるし、TCBのサイズが圧縮
できるではないかというのも確かに理屈では
あります。今回は後者の方法を採用してみ



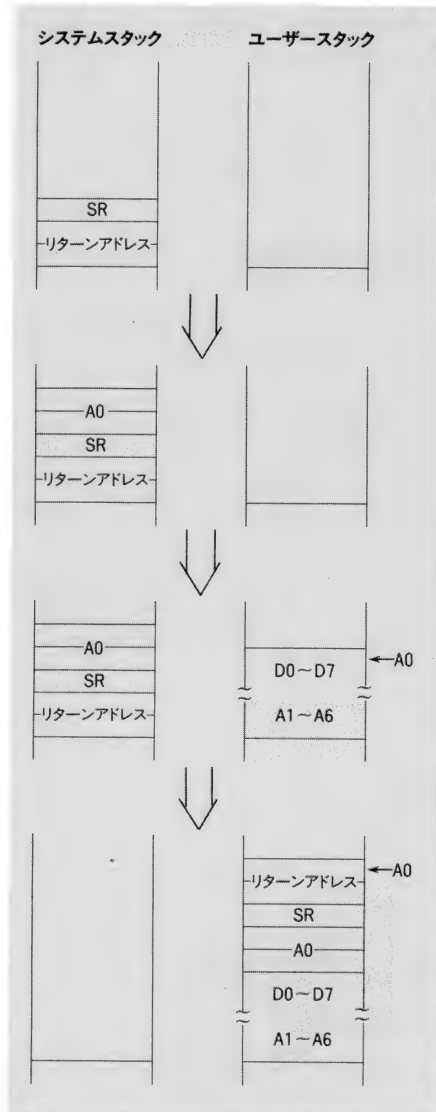
ました。スタックのオーバーフローなどを考えると若干前者に分があるかなという感じもしないではないのですが……。

まず、割り込みが入った時点で一時的にワークとして使うレジスタをシステムスタックにセーブします。

ここではA0がそれにあたります。次にA0にユーザースタックの内容をコピーし、ユーザースタック領域をA0で指し示します。ここで、残るレジスタD0~D7, A1~A6をmovem命令を使って一気に積み上げていきます。さらに、先ほどシステムスタックに積み上げておいたA0の値と、割り込み発生時にCPUによって自動的に積まれていたステータスレジスタと帰りの番地を積み上げれば、全レジスタのセーブが完了します。このときのスタックの動きを図5に書いておいたので、参考にしてください。

ここで、ステータスレジスタのチェックが入っているのは、X68000のIOCS コール

図5 割り込み発生後のスタックの動き



リスト1

```

===== TSS.S =====
1: *
2: * 時分割処理サンプルプログラム
3: *
4: *
5: * 1990-05-24 Written by M.Kuwano
6: * No rights reserved.
7: *
8: .globl _task_1,_task_2,_task_3
9: .xref _tss0,_ts_init
10: .include iocscall.mac
11: .include doscall.mac
12: FLAG_SUPER equ 5 * スーパーバイザか否かを示すフラグ
13:
14: .text
15: sema: dc.b 0
16:
17: .even
18:
19: *
20: * 割り込み処理
21: *
22: tss: move.l a0,-(sp) * スーパーバイザスタックを拝借
23: move.l USP,a0 * ユーザスタックの取り出し
24: movem.l d0-d7/a1-a6,-(a0) * a0以外をユーザスタックに待避
25: move.l (sp)+,-(a0) * a0も待避
26: move.w (sp)+,-(a0) * フラグ
27: move.l (sp)+,-(a0) * リターン・アドレス
28: bsr chk_flag * フラグをチェック
29: bne tss_pass * 既にスーパーバイザだったらパス
30: move.l a0,-(sp) * 引き数としてA0の値を渡す
31: bsr tss0 * スケジューリング
32: addq.l #4,sp * 値はD0に入ってくる
33: move.l d0,a0
34: tss_pass:
35: move.l (a0)+,-(sp) * 帰りを積んで
36: move.w (a0)+,-(sp) * フラグも積んで
37: move.l (a0)+,-(sp) * a0を一時待避
38: movem.l (a0)+,d0-d7/a1-a6 * レジスタ復帰
39: move.l a0,USP * ユーザスタックのすりかえ
40: move.l (sp)+,a0 * a0も復帰
41: rte
42:
43: *
44: *
45: * タスク0~3
46: *
47: _task_0: bsr _ts_init
48: lea.l tss,a1
49: move.l #57,d1
50: IOCS _VDISPST * 垂直同期の割り込みを使う
51: loop:
52: move.w #531,d1
53: bsr get_sema
54: IOCS _B_PUTC
55: bsr put_sema
56: bsr wait
57: bra loop
58:
59: _task_1: move.w #532,d1
60: bsr get_sema
61: IOCS _B_PUTC
62: bsr put_sema
63: bsr wait
64: bra _task_1
65:
66: _task_2: move.w #533,d1
67: bsr get_sema
68: IOCS _B_PUTC
69: bsr put_sema
70: bsr wait
71: bra _task_2
72:
73: _task_3: move.w #534,d1
74: bsr get_sema
75: IOCS _B_PUTC
76: bsr put_sema
77: bsr wait
78: bra _task_3
79:
80: *
81: * 時間待ちループ
82: *
83: wait:
84: move.w #0xffff,d0
85: dummy:
86: dbra d0,dummy
87: rts
88:
89: *
90: * 既にスーパーバイザモードになっていたかのチェック
91: *
92: chk_flag: btst.b #FLAG_SUPER,4(A0)
93: rts
94:
95: *
96: * セマフォ獲得 取れるまで無限ループ
97: *
98: get_sema: bset.b #1,sema
99: bne get_sema
100: rts
101:
102: *
103: * セマフォ解放
104: *
105: put_sema: bclr.b #1,sema
106: rts
107:
108: .end _task_0

```


やDOSコールの中でも割り込みが入ってしまうためです。すべての割り込みを自分のプログラムで管理しているのなら、ここでは割り込みがあったというフラグだけを立てて、もともと実行していたIOCSコールなどの処理を行い、最後にいまの割り込みの処理をしておくといったテクニックも使えるのですが、横取りした割り込み以外はすべてHumanでセットされたままで使う関係上しかたがないので、すでにスーパーバイザモードにいるようであれば、なにもしずに元に戻るようになっています。

レジスタの待避が終わったらA0の値をもってTSS0というサブルーチン呼び出します。TSS0はわかりやすいようにCで書いてみました。もちろんアセンブラで書くこともそれほど難しくはありませんが、見やすさという点ではCのほうがよいでしょう。TSS0では動かしていたタスクの番号からTCBのアドレスを計算し、その中に渡されたA0の位置を保存します。

あれ？ USP（ユーザースタックポインタ）は？と思われた方も多いでしょう。アセンブラのほうでTSS0を呼び出したあとからRTE命令までの部分を読んでみてください。最後に積み上げた場所から順次データを引き抜いていくって、最後に残ったA0の値、これは元のスタックの位置にほかなりません。ここで、A0をUSPに移し、さらにA0自身の値も元に戻せばめでたしめでたしとなるわけです。このときのスタックの動きは、ちょうど割り込みが入ったときと逆になります。

次に各タスクの選択のやり方を見てみることにしましょう。0番のタスクは起動したときから動いていたプログラム、その他のものは順番がくるのを待って1列に並んでいます（図6（a）参照）。この先頭を指しているのがrdy_q_head、最後を指しているのがrdy_q_tailです。

現在実行しているタスクはc_taskで指し示されるTCBのものです。当然、c_taskの初期値はTCB[0]を指すようになっています。ここで割り込みが入ると、TCBのつなぎ換えが行われます。いままで実行されていたタスク（c_taskの指しているTCBのタスク）のTCBはrdy_q_tailが示すTCBの後ろにつながれ、代わりにこれまでrdy_q_headが指していた、キューの先頭のTCBをc_taskが指すようになります。rdy_q_headは、いままでキューの2番目にいたTCBを指すように変更されます（図6（b）参照）。

アセンブラプログラム中のget_sema, put_semaの2つは、シングルタスクでのプロ

グラムしか見慣れない方にはちょっと珍しいコーディングかもしれませんが、これは、セマフォと呼ばれるプログラミング手法のひとつで適当なプログラム領域の先頭と末尾でget_semaとput_semaを実行することで、これに挟まれた区間は一度には必ずひとつのプログラムしか実行しないようにするしかけです。

get_semaで使われているbset命令は、単

に指定されたビット位置を1にするだけでなく、そのビット位置の前回値が0であったか否かによってZフラグが変化します。semaの領域は初期値として0が入っています。いま、タスクAがここに飛んできたしましょう。bset命令によって、semaが1、Zフラグはもともと0でしたから1になります。Zフラグが立っていますから、そのまま下に向かい、IOCSコールが行えます。

リスト2

```
===== TSC.C =====

1: /*
2: *      時分割処理サポートルーチン
3: *      T C B 管理 & 初期化
4: *
5: *      1990-05-24  Written by M.kuwano
6: *      No rights reserved.
7: */
8:
9: #define  READY          0
10: #define  EOQ            -1 /* キューの最後を示す */
11: #define  NR_TASKS       4 /* タスクの数 */
12: #define  TASK_STK_SIZE  0x100 /* タスクのスタック */
13:
14: extern void task_1(), task_2(), task_3();
15:
16: struct TCB {
17:     unsigned char state; /* タスクステート（今回は使っていない） */
18:     unsigned char sstate; /* サブ・ステート（上に同じ） */
19:     unsigned int *usp; /* ユーザー・スタック */
20:     struct TCB *tcb_link; /* レディキューのリンク */
21: } tcb[NR_TASKS];
22:
23: struct STACK {
24:     unsigned int task_sp[TASK_STK_SIZE];
25:     void *task_pc;
26:     unsigned short task_sr;
27:     unsigned int task_regs[16];
28: } t_stack[NR_TASKS];
29:
30: struct TCB *rdy_q_head; /* レディ・キューの先頭 */
31: struct TCB *rdy_q_tail; /* レディ・キューの末尾 */
32: struct TCB *c_task; /* カレントタスク */
33:
34: void ts_init()
35: {
36:     unsigned int i;
37:     struct TCB *t;
38:     struct STACK *s;
39:     t_stack[1].task_pc = task_1;
40:     t_stack[2].task_pc = task_2;
41:     t_stack[3].task_pc = task_3;
42:     c_task = tcb;
43:     c_task->state = READY;
44:     c_task->sstate = 0;
45:     c_task->tcb_link = EOQ;
46:     t = rdy_q_head = &tcb[1];
47:     s = &t_stack[1];
48:     for (i=1; i<NR_TASKS-1; i++, t++, s++) {
49:         t->state = READY;
50:         t->sstate = 0;
51:         t->usp = &s->task_pc;
52:         t->tcb_link = &tcb[i+1];
53:         s->task_sr = 0;
54:     }
55:     t->state = READY;
56:     t->sstate = 0;
57:     t->usp = &t_stack[i].task_pc;
58:     t->tcb_link = EOQ;
59:     s->task_sr = 0;
60:     rdy_q_tail = t;
61: }
62:
63: void *tss0(stack)
64:     unsigned int *stack;
65: {
66:     struct TCB *p;
67:     c_task->usp = stack; /* スタックをTCBにしまう */
68:     if (rdy_q_head != EOQ) {
69:         c_task->tcb_link = EOQ; /* 自分が最後になる */
70:         rdy_q_tail->tcb_link = c_task; /* キューの最後につなげる */
71:         rdy_q_tail = c_task; /* キューのおしりは自分 */
72:         c_task = rdy_q_head; /* 次の方、どーぞ */
73:         rdy_q_head = c_task->tcb_link; /* 先頭をキューからはずす */
74:         c_task->tcb_link = EOQ; /* 気分の問題 */
75:     }
76:     return(c_task->usp);
77: }
```


これより少し遅れて次のタスクBがここ
にきたとします。同じようにbset 命
令を使
うと、今度はすでに1が立っていますか
らZフラグが立ちません。したがって、こ
こでbsetをやり続けることになるわけ
です。

さて、タスクAが処理を終わり、put_se
maを実行したとします。ここでsemaの値
は0に戻ります。そして、次にタスクBが
get_semaをやると今度はZフラグが立つ
ので、無事にループが止まり、下に抜けて
くるわけです。

このようなことが行えるのは、割り込み
というのが（たとえNMIであろうと）、1命
令の処理の途中では入ってこれないためな
のです。bset 命令はメモリからのデータの
読み込みのあと、再び書き込むという、2
段階で実行されますが、この隙間には絶対
割り込みは入らないのです。もし、ここを
Cなどで書くと、おそらくmove 命令が複
数個並ぶことになるでしょう。こうなると、
セマフォの役をはたさなくなってしまう
す。

もし、タスクAがsemaの値を持ってきた
ところでタスクBに切り替わり、ここでも
タスクBがsemaを読み出したとします。タ
スクA、Bの両方ともsemaの値としては0
を読み出すことになるため、セマフォが取
れたつもりになってしまいます。

8086なら、ここで一時的に割り込みを禁
止するなどという、野蛮な手法も使えるの
ですが、68000では割り込みフラグの操作は
特権命令ですから、スーパーバイザモード
でなくては使えません。各タスクはユーザ
ーモードで動いていますから、もしこんな
ことをすればただちに特権命令違反になっ
てしまいます。

このようなクリティカルな部分が記述で
きるのはやはりアセンブラならではとい
えるでしょう。

動かしてみよう

サンプルのソースを入力してコンパイル、
アセンブル、リンクしてください（cc tss.
s tsc.c とでもすればよいでしょう）。エラ
ーメッセージが出なくても、打ち間違えや
行の抜けなどがあるとあっさり暴走して
くれますので、外付けならば実行前にハー
ドディスクの電源はOFF（念のため、ブ
レイクキーを押してからにしましょう）、フ
ロッピーディスクも抜いておいたほうが賢
明でしょう。

うまくいけば、1から4の数字（タスク
の番号と一致しています）がゾロゾロゾ

……と表示されるはずですが。タスク0から
は緑もゆかりもないまったく別のところ
にあるプログラムがちゃんと同時に（正確
には順番にですが）動いているのがわか
ることでしょう。ブレイクキーセンスな
どはやっていませんので、やめなくなっ
たら、リセットしてください。

うまく動いたようでしたら、いろいろと
改造してみるとよいでしょう。X68000の
場合、グラフィックなどもIOCSでサポー
トされていますから、マルチタスクでグ
ラフィック画面にいろいろ表示させる
というのも簡単なことです。

おわりに

アセンブラでのプログラミングのサンプ
ルというと文字表示やら簡単な計算やら
が多いのですが、今回はアセンブラなら
ではということを考え、68000の割り
込み処理をつつき、簡単な時分割処理
を行ってみました。

単なる時分割ですから、タスク間の通
信

も、ファイルやメモリの管理もありません。
タスクの状態遷移という考えすらないごく
原始的なものではありますが、実のところ、
巨大なマルチタスクOSもそのカーネル部
分まで下がってみると、最後はこの程度
のレベルに落ちつくのです。

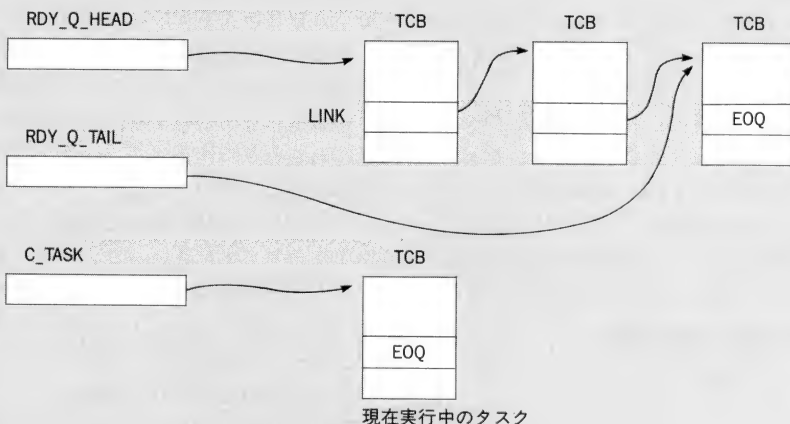
要するにどんな巨大なOSであっても、最
下層のところでは複数のプログラムの状
態を記録しておいて、割り込み（エクセ
プションといったほうが正しいかもしれ
ませんが）が入ったときにそれらの中に
現在実行していたタスクの状態を記録し、
次に動作させるタスクの記録を引っ張り
出してきてその実行にとりかかるとい
うだけのことなのです。

大胆不敵にいってしまうと、この段階
がわかってしまえばマルチタスクの気分
が味わえる程度のOS（モドキ）ならそれ
ほど苦勞しなくても作れてしまうのでは
ないでしょうか。

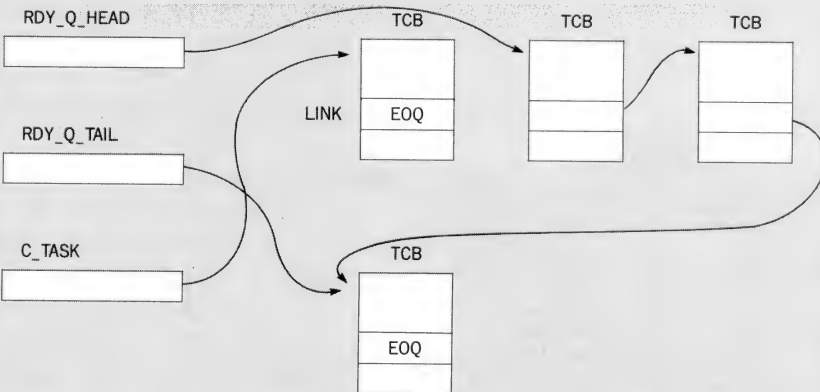
いっそ、夢はでっかくUNIXライクな
パーソナルOSを作ろう会！……誰かや
る人いませんか？

図6 タスク切り替えの考え方

(a) タスク切り替え前の状態



(b) タスク切り替え後の状態



S-OS&REDAを忘れていませんか

Yamada Junji
山田 純二

X1やMZユーザーの皆さん、お元気ですか。大好きなマーベルランドのCDを買ったら、予約もしていないのにポスターをもらってウキウキしている山田純二です。

さて、6月号発表のアンケート結果を見ると、X1、MZユーザーの40%も的人がアセンブラを使っているそうです。自力でマシンをコントロールできれば8ビットマシンでもいろいろと楽しめますからね。とはいえ、マシン語をやろうと思い立ってS-OS“SWORD”用のアセンブラREDAやZEDAなどを打ち込んだけど、結局投げ出してしまったという人も多いでしょう。今回は、これらの使い方について、もう一度だけ基礎の基礎から説明していくつもりです。いったんあきらめてしまった人も、実はこれからと思っている人も、挑戦してみてください。もうアセンブラは、マスターしたという人も、昔を懐かしみながら読んでみてもらえると、僕はとてもうれしいです。

使用システムはS-OS“SWORD”，アセンブラは、REDAを使います。ほかに、OHM-Z80などや、今月号のWZDなどがありますが、多機能すぎて初心者向きではな

い（と僕は思った）ので、今回は、REDAを使って話を進めていこうと思います。

超入門編

●STEP1 モニタでなにをするのか

まず、“SWORD”を起動すると、プロンプト“#”が表示されます。これがS-OSのモニタと呼ばれるコマンドモードで、基本的なファイル操作を行ったり、プログラムの実行をさせます。S-OSのモニタに対するコマンドはすべて、この“#”の後ろに続けて書かなければなりません¹⁾。

●STEP2 モニタサブルーチンとは

多機種にわたる“SWORD”上で、アプリケーションを作成するには、機種の違いをなくすため用意された、共通ルーチンを使う必要があります。システムの上下関係は図1のようになっています。これを見れば、なぜ、わざわざモニタサブルーチンを使わなければならないのかおわかりでしょう。

サブルーチンの具体的な使用例として、リスト1を見てください。ここでは、ディスクからの連続セクタリードを行うサブルーチンをコールしています。まず、サブルーチンコールに必要なパラメータとして、

A……読み込むレコード数

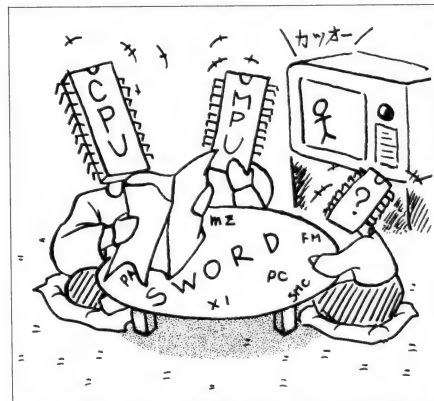
DE……読み込む先頭レコードナンバー

HL……読み込み先のアドレス

をそれぞれ指定されたレジスタへ渡してから、2000Hのアドレスをコールします。サブルーチンコールは、すべてこのようにして使います。それぞれのサブルーチンによって破壊されるレジスタが違うので、必要に応じてレジスタを保存しましょう。

●STEP3 いよいよアセンブラ

それでは、実際にリスト1をアセンブルしてみましょう。初めに、モニタ上からREDAを読み込み、J3000で起動させます。最初は、アセンブラモードで立ち上がりますから、“E”コマンドでエディタのコマンドモードに移ります。ここでは、テキストの初期化、テキストファイルのセーブ/ロード



などが行えます。新しくテキストを入力する場合は、“N”コマンドによって、テキストエリアを初期化しておいたほうが安全です。

テキストの編集は“E”コマンドで、エディタモードへ移ってから、リスト1を入力しましょう。入力が終わったならば、いったんデバイス（ディスク）にセーブしてから、シフト+ブレイクでアセンブラモードへ戻り、

:A/

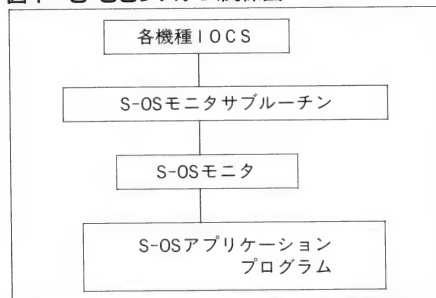
とすれば、アセンブルできます。“A”コマンドの後ろのスラッシュは、アセンブル時に、画面へアセンブルリストを出力させるものです。慣れないうちは、リストを表示させながらアセンブルしたほうがいいでしょう。なぜかって？ それは、オブジェクトとソースリストの関係がよくわかるからです。

アセンブルエラーが出ると、自動的にエディタモードへ移ってエラーの出た行を表示してくれるので、元のリストとよく見比べて修正後、再度アセンブル。無事アセンブルが終了すると、使われたラベル数と、スタートアドレス、エンドアドレス、エントリアドレス、オフセットアドレス、を表示して完了です。

実行は、アセンブラモードか、“SWORD”

1) モニタコマンドについては、“SWORD”掲載号の記事を見れば一覧表が載っています（たとえば、先月号の125ページ）。

図1 S-OSシステム関係図



リスト1

```

1
2 :SAMPLE1
3 #DRDSB EQU $2000
4
5          ORG $9000
6 HAJIME
7 LD A,01
8 LD HL,INPUT
9 LD DE,(NO.)
10 CALL #DRDSB
11 RET
12 INPUT DS 256
13 NO. DW 25
    
```


のモニタ上から、J9000です。

●STEP4 ソースリストの構成と疑似命令

では、ひととおりアセンブラの扱い方がわかったところで、ソースリストがどのように記述されているか説明していきましょう。これがわかっていなければ、ソースリストは、読めませんからね。

ステートメントは、大きく分けて3つの部分に分類でき、左から、ラベル、オペコード、オペランドから構成されています。

まず、ラベルですが、これは必ず文頭に書かなければなりません。ひとつでも、空白があるとアセンブラはそれをオペコードと解釈してしまいます。オペコードには、コンピュータに対する命令を書きます。オペランドとは、オペコードで書いた命令を、詳しく規定するものです。オペコードとオペランドの間は、ひとつ以上の空白が必要です。

これら3つの区切りは厳密に規定されてはいませんが、リストの保守性を高めるためにも、自分なりに綺麗に区切っておくといいでしょう。もうひとつ、“;”の書かれたあとは、行末まで、コメント行として解釈されます。

さて、ソースリストを読むためにはもうひとつ、疑似命令というものも知っておかなければなりません。疑似命令というのはアセンブラに対する命令で、プログラムの実行には関係はありませんが、プログラムを作るときには必要不可欠なので、覚えておきましょう。

●ORG命令

アセンブラが生成する、オブジェクトプログラムの、先頭アドレスを指示する命令です。

●ラベル

BASICで使うラベルと同じようなものですが、アセンブラでは、単に目印となるだけでなく、ラベルにプログラムのアドレスが定義される、という意味があります。

●EQU命令

ラベルにオペコードの値を定義する命令。値は必ず2バイトで、指定しなければなりません。

●データ定義命令

DEFB 1バイトのデータを格納

DEFW 2バイトのデータを格納

DEFS オペランドで指定した数だけ
メモリの領域を確保

DEFM オペランドに指定した、文字
列をメモリに格納

以上の4つがもっとも重要な疑似命令です。

さあ、実践だ！

それでは、モニタサブルーチンの使用法の説明もかねて、ちょっと役に立つ可愛いサブルーチンを作っていきます。“SWORD”で扱えるファイルには、テキストファイルとオブジェクトファイルの2種類ありますが、モニタ上では、テキストファイルが扱えません²⁾。ここはやっぱり、扱えないのなら、自分で扱えるようにしてしまえの精神で、外部コマンドとしてのTYPEコマンドを実際に作ってみましょう。

●メインルーチンの作成

まず、リスト2 (サブルーチン名BEGIN) の説明です。これは、起動したときのメッセージ表示と、のぞきたいファイル名を入力するルーチンです。メッセージの表示方法としては、ここで使っている#MPRINTのほかには#MSG、#MSXなどがあります。

#MPRINTというのは、コールした次のアドレスから00Hがあるまで、ASCIIコードとみなして文字列出力します。#MSGはDEレジスタに示すアドレスから、0DHがあるまでASCIIコードとみなして文字列出力。#MSXは、#MSGと大体同じで文字列のエンドコードが00Hです。

3つのサブルーチンに共通するのは、文字列の表示後に改行しないということです。改行させたい場合は、表示したあとに#LTNLをコールするか、文字列中に改行コードの0DHを入れておく必要があります。

で、お次はファイル名の入力。これは、キーボードからの1行入力ルーチンを使います。DEレジスタにキー入力バッファの先頭アドレスを入れてからコール。バッファは、最低でも80バイト必要ですので、自分でバッファを用意するか、各機種のモニタに用意されているバッファを使いましょう。

各機種用のバッファアドレスは、S-OSのワークエリアの#KBFADに格納されています。注意しなければならないのは、画面の左端からバッファに取り込まれることです。メッセージを表示して文字列入力したときには、入力後に、メッセージ分の文字列をスキップさせてやらなくてはなりません。これは、出力されてくるDEレジスタはバッファの先頭アドレスを示しているからです。リスト2の場合、メッセージは10文字使っているの、DEレジスタに10文字分足してから、ファイル入力ルーチンへ渡しているのがわかるでしょう。

●ファイルの入力

S-OSでファイルの入力をする場合は、以下に示す手順で行います。ファイル入力ルーチンはリスト3、サブルーチン名はFILE READです。これを見ながら読んでみてください³⁾。

1) ファイル名の入っている先頭アドレスをDEレジスタにセット、Aレジスタにファイルのアトリビュートをセットしてから、#FILEをコール。

2) 次に、ファイルの情報を取り出すために、#ROPENをコール。

3) #ROPENで取り出したファイル情報に従って、デバイス上のファイルを読み込む。これは#RDDをコールします。ちゃんと#ROPENをコールしてファイルをオープンしないとエラーとなってしまうので注意してください。

4) おわり。

1)のファイルのアトリビュートとは、読み込むファイルの属性を指定するもので、オブジェクトファイルの場合は1、テキストファイルの場合は4と指定します。この場合はテキストファイルを扱うので、Aレジスタに4をセットして#FILEをコールしています。そして、#ROPENでファイル情報の取り出しを行い、その後、強制的にファイルのロード先を変更しています。

ロード先の変更は#ROPEN後、#DTADRを変更すれば、ファイルは新しく指定されたアドレスにロードされます。E-MATEなどはファイルのロード先をオフセットでセーブしているため、そのままだと0000Hにテキストをロードしてしまうので、#ROPEN後に変更してやらねばなりません。エラーがあるとキャリフラグをセットして帰ってくるので、そのときは#ERRORにジャンプさせるようにしておきましょう。

●テキストの表示ルーチン

ファイル入力も終わり、次はリスト4 (サブルーチン名TYPEMAIN) の読み込んだテキスト表示の説明です。テキストの構造は、文字列、改行コード0DH、の順番に1行ごとに並んでいて、エンドコードは00Hとなっています。これを知っていれば、1行ご

2) テキストファイルはアプリケーション用の文字列ファイル、オブジェクトファイルは実行可能なマシン語プログラム、またはマシン語データとなっています。

3) このTYPEルーチンは、一度メモリに読み込んでから、ドバーッと表示するというあまり賢くない構造です。本来なら1クラスごとに読み込んで、そのつど表示していくのが正しい姿ですが、根性のある人は、自力で挑戦してみてください。



とに改行して、エンドコードを調べながら、テキストを文字列表示する必要はなく、ドパーッと文字表示を行ってあげれば、1行ごとに自動的に改行してくれるのがわかるでしょう。

1 文字表示は、AレジスタにASCIIコードをセットして#PRINTルーチンをコールすれば、1文字ポコンと表示されます。表示をやりっぱなしだと、ちょっと困ったちゃんなので、#PAUSEを使ってテキスト表示の一時停止機能をつけてあります。#PAUSEの機能は、スペースキーが押されたら何かキーを押すまでリターンせず、シフト+ブレイクが押されると、コールされた次の2バイトに示されるアドレスにジャンプします。この場合はブレイクを押されたら表示を終了させたいので、TM3をセットしてあります。

以上で、リストの解説はおしまい。それぞれのリストをBEGIN, READ, PRINTのファイル名でセーブし、アセンブルするときは、REDAのアセンブルモードから

:ABEGIN:READ:PRINT

と打ち込むとアセンブルできます。実行は、リスト1のサンプルと同じ方法で行ってください。

リスト2

```

1
2 ;TYPE COMMAND Ver 1.0
3
4 #MPRINT EQU $1FE2
5 #GETL EQU $1FD3
6 #ERROR EQU $2033
7
8 ORG $9000
9 BEGIN
10 CALL #MPRINT
11 DB "*** TYPE Ver 1.0 ***"
12 DB $0D,00
13 CALL #MPRINT
14 DB "File Name:",00
15 CALL #GETL
16 LD A,(DE)
17 CP $1B
18 RET Z
19 LD HL,10
20 ADD HL,DE
21 EX DE,HL
22 CALL FILEREAD
23 JP C,#ERROR
24 CALL TYPEMAIN
25 RET
26 BEGINEND

```

●オフセットをつけてみよう

リストをそのまま打ち込んで、アセンブルすると9000Hにオブジェクトが生成されますが、メモリのど真ん中にプログラムを置くのはまいち気持ち悪いので、今度は、3000Hからプログラムを生成させるようにしてみましょう。

これは、素直にORG命令のところを3000Hに変更しただけでいいように思いますが、3000Hにはアセンブラが、でーんと居座っているの、ORG命令で、3000Hにしてしまうとアセンブラが破壊されてしまいます。これを避けるには、3000Hから動作するプログラムを、一時的にシステムと重ならないような別の場所にアセンブルしてやらなければなりません。そのための疑似命令として、OFFSET命令があります。使用法は、

OFFSET \$6000

ORG \$3000

のようにして使います。ORG命令で指定したアドレス+OFFSET命令で指定したアドレスにオブジェクトを生成することができます。この場合だと6000H+3000H=9000Hのアドレスに、3000Hで動作するプログラムが出来上がります。

さて、このプログラムを実行するために

自分でやっちゃえ!

ノーマルのまま“SWORD”のモニタを使っている、慣れるにしたがって、いろいろと不満が出てくることでしょう。そんなときには、このように外部コマンドとして、どんどん自分でシステムを拡張させていくといいと思います。内部コマンドの増設は、メモリの関係上難しいものがありますが、このようにしていけば、ディスクの容量があるかぎりコマンドが作れますからね。

ちなみに僕は、アセンブラで開発をすることが多く、いちいちエディタやアセンブラをロードするのはめんどくさかったので、モニタ上から

は、9000Hから生成されたオブジェクトを、ORG命令で指定したアドレスにロードしなおす必要がありますが、REDAで分割アセンブルする場合は、最後のファイルセーブで自動的にこれらの処理をやってくれるので、あまり気にしなくていいかもしれません。というところで、サンプルのプログラムの変更点は、リスト2のBEGINの8行目に、

OFFSET \$6000

の1行をつけ加えて、9行目のORG命令のアドレスを3000Hに変更してからアセンブルしてください。アセンブルが終わると、セーブするかどうか聞いてきますので、“Y”と入力してセーブしておきましょう。この場合、ファイル名は、

BEGIN.OBJ

で強制的にセーブされてしまいます。実行は、いったんBEGIN.OBJをS-OSのモニタからロードしなおして、

#J3000

で実行してください。

さて、実例をまじえてアセンブラとS-OS“SWORD”の使い方を説明してきました。いかがだったでしょうか。まだ8ビットで頑張ろうという人はぜひともマシン語を覚えて愛機を活用してください。

リスト3

```

1
2 ;FILE READ ROUTINE
3
4 #RDD EQU $1FA6
5 #FILE EQU $1FA3
6 #ROPEN EQU $2009
7 #DTADR EQU $1F70
8
9 ORG BEGINEND
10
11 FILEREAD
12 LD A,04
13 CALL #FILE
14 FR2
15 CALL #ROPEN
16 RET C
17 LD HL,MAINEND
18 LD (#DTADR),HL
19 CALL #RDD
20 RET
21 READEND
22
(ディスク専用です)

```

リスト4

```

1
2 ;TEXT PRINT ROUTINE
3
4 #PRINT EQU $1FF4
5 #LTNL EQU $1FEE
6 #PAUSE EQU $1FC7
7
8 ORG READEND
9
10 TYPEMAIN
11 LD HL,MAINEND
12 TM2
13 LD A,(HL)
14 OR A
15 JR Z,TM3
16 INC HL
17 CALL #PRINT
18 CALL #PAUSE
19 DW TM3
20 JR TM2
21 TM3
22 CALL #LTNL
23 RET
24 MAINEND

```


●リロケータブルアセンブラ登場

さて、すでに先月の「リロケータブルフォーマットの取り決め」で概要をお知らせしたとおり、S-OS上でリロケータブルファイルを扱うためのアセンブラ、WZDを発表します。予告ではこのアセンブラが出力したファイルをリンクするリンク、WLKも同時発表となっていました、ページ数の都合によりアセンブラ部分のみの発表となりました。あらかじめご了承ください。

今回のWZDはエディタで書かれたソースプログラムをリロケータブルファイルに変換するためのものです。これをさらに実行形式にするには来月掲載予定のWLKが必要です。ページが許せば専用のライブラリアン、WLBも来月（以降）で紹介することになります。

以前から何度か紹介しているように、これらのシステムはもともS-OS上でC言語を効率よく運用するために開発されたものです（もちろんC言語を動かす以外にもおいしい使い方はたくさん考えられますが）。C言語が加わればS-OSの広がりもさらに大きくなるでしょう。S-OS上にどのようにしてC言語を持ってくるかは、また今後のお楽しみということにしておきましょう。

第96部

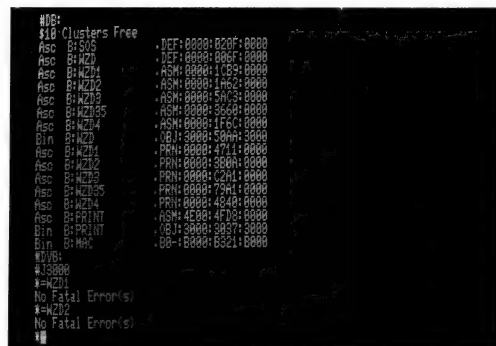
リロケータブルアセンブラ WZD

●リロケータブルファイルの扱い

今回のWZDを使用する際には必ずフロッピーディスクまたはRAMディスクが必要になってきます。これまでS-OSの世界ではディスクがないと使えないアプリケーションというのはごく限られた存在でした。ディスクエディタや変身セットの一部の機能、SLANGなどのインクルード機能、ファイル入出力ライブラリといったところでしょうか。

S-OSシステムはカセットテープでもディスクでも同様のファイルを扱うというポリシーを堅持していても、システムに対する要求が次第に高度なものになってくると、どうしても1文字単位の入出力やリロケータブルファイルのリンクなどといったカセットテープの不得意な問題が浮かび上がってきます。編集部でもそれらをカセットテープでも変わらずに実現するための方法をいろいろ論議したものでした。カセットベースでのダイナミックリンクの可能性などは非常に興味深いテーマです。しかし、いつも結論は「技術的に不可能ではないが、かなり不毛」というところに落ち着きます。

今回のシステムはS-OSにおけるひとつの岐路といえるかもしれません。



●S-OSの系譜(11)

パソコンユーザーにとってもっとも親しみのあるプログラミング言語、それはやはりBASICでしょう。1986年9月、ついにS-OS上にBASIC言語が登場しました。このFuzzyBASICとはいったいかなるもののでしょうか。単に「パソコンユーザーの作った整数型BASIC」だと思っていると大間違いです。

ブロックIFなどの構造化命令は当然として、ユーザー定義手続き、ユーザー定義関数（もちろん局所変数つき）、I/O配列、ユーザースタック、HuBASICでお馴染みのKEY0命令、はてはブロック転送命令まで備えています。BASICという言語の持つ泥くさいイメージとは違い、洗練された高級言語としてプログラミングユーザーに受け入れられました。

確かにプログラミングに適した言語仕様、省略形のサポートやシャープ、マイクロソフト系両方に対応したコマンドなどのユーザーインタフェース部分も備えたフレンドリな面、かなり高速なうえ、使う側が心得ていれば、高速アクセス変数、マシン語に密着した命令群を使ってさらに最適化することができるなどのマニアックな面も魅力ですが、BASICという言語にこだわりつつも、従来のパソコン用BASICを超える発想とこだわり方、そしてユニークさがなにより魅力的だったのではないのでしょうか。

その後、FuzzyBASICはグラフィックのサポート、命令の強化などを加え、やがて現れるFuzzyBASICコンパイラによってさらに完成度を高めていくことになります。

また、直接全機種共通システムとは関係ありませんが、1986年9月号は高速グラフィックパッケージMAGICという弩級プログラムも掲載され、両プログラムのためにいまでは過去最高の希少バックナンバーとして知られています。MAGICはその高性能もさることながら、グラフィック共通化の鍵として以後S-OSシステムと密接な関係を持ちつつ、独自の世界を広げていきました。

リロケータブルアセンブラ WZD

Ishigami Tatsuya

石上 達也

6月号で予告したリロケータブルアセンブラ, WZDの登場です。リロケータブルフォーマットによって S-OS の新しい可能性が広がっていきます。FDDがないと使用できませんので注意が必要です。なお, WZDは“ダブルゼーダ”と読んでください。

先月号の予告ではリンクとともに発表することになっていましたが、ページの都合で、リンクの発表は来月号で、ということにさせていただきます。とりあえず、リロケータブルアセンブラ WZD (ダブル・ゼーダと読んでください) を発表させていただきます。それから先月号では触れるのを忘れてしまいましたが、このWZDはフロッピーディスクがないと動作しません。

このプログラムは、いつかも予告したようにS-OS上でCコンパイラを走らせるために開発しましたが、単体でも十分実用的だと信じています。ちなみに、WZDとWLKの原形はCP/M上で作りましたが、そこからはWZD自身を使って開発しました。来月号(?)で発表予定のWLBは90%以上WZDを使って作成しました。

私の学部1年のはとんどの時間をかけて作ったプログラムです。なるべく多くの人に使ってもらいたいののでこのプログラムはPDSとします (どっかのネットに入れた場合は一応私に知らせてください)。

では使いなんぞを

WZDの起動の仕方には、2種類あります。S-OSのコマンドラインからパラメータを指定する方法と、WZD内のコマンドラインから、パラメータを指定する方法です。

●S-OSのコマンドラインから

まず、S-OSの拡張をしている方は、

WZD パラメータ

とすると動作を開始します。アセンブルが終わると#を示して、S-OSのコマンドラインに戻ります。

次に、S-OSの拡張をされていない方は、

LWZD

で、WZDをディスクから読み込んで、

J3000 パラメータ

とすると動作を開始します。あとは拡張をしているときと同じです。この方法は、1本のプログラムをアセンブルしたいときに

用いると便利です。

●WZDを起動してから

LWZD

J3000

でWZDを起動すると入力を促すプロンプト‘*’を表示してWZDのコマンドラインに入ります。‘*’の後ろに続けてパラメータを与えます。パラメータによって指示された動作が終了すると、再び入力を促すプロンプト‘*’を表示して、WZDのコマンドラインに入ります。

さらに、別のプログラムをアセンブルしたい場合には、1回目と同じようにパラメータを与え、WZDを終了したいときには、シフト+ブレイクキーを押すとS-OSのコマンドラインに戻ります。この方法は、アセンブルしたいプログラムが複数あるときに便利です。

パラメータといたって

基本的には、MACRO-80に準じています。以下に、WZDが起動してからパラメータを指定した場合について具体的な使い方の例を挙げておきます。

パラメータの指定で一番オーソドックスなのは、

*[リスト・ファイル名],[リロケータブルファイル名]=[ソースファイル名]のかたちです。

で、たいいては、[リロケータブルファイル名]と[ソースファイル名]は拡張子が違うだけであとは同じ、[リストファイル名]は無指定 (つまりリストファイルは作らない) にしたいわけで、そんなときには、

*=[ファイル名]

とすれば、タイプ数も少なくすみます。やはり、[リストファイル]も同じ名前ではいいというときには

*=[ファイル名]/L

です。

ソースファイルはBドライブであって、

リロケータブルファイルはAドライブに作りたいときは、

*A:[リロケータブルファイル名]=
B:[ソースファイル名]

です。さらに詳しくはリファレンスマニュアルを見てください。

プログラムについて

ニーモニックからマシン語への変換については、1989年2月号掲載のREDAのルーチンをほぼそのまま使わせていただいています。ただREDAのルーチンは、裏レジスタまでも駆使し、テキストポインタにDEレジスタを用いたり、かなりスピードを重視したルーチンが多かったのですが、WZDではスピードはあまり重視せずに、安全性を重視して一部書き直してあります。

さらに、ラベルの検出方法をREDAのオープンハッシュ法からチェーンハッシュ法に切り替えています。別に深い意味はありませんが、心持ち速くなったような気がします。ハッシュ法とはなにかを説明しよう

使用上の注意

EQU文のパラメータは1パス目のその時点において、値が確定している必要があります。このことによって一部、REDAのソースファイルがアセンブルできないことがあるかもしれませんが、そのときは値が確定する順にラベルを並べ替えてください。

PHASE文とDEPHASE文のあいだにおいてセグメントのモードを切り替えてはいけません。また、PHASE文とDEPHASE文のあいだではアドレス情報がアブソリュート値で処理されます。

インクルードファイルのネスティングは4重までです。呼ばれたファイルから呼び出したファイルを、呼び出しては絶対にいけません (無限ループに陥ります)。

リストファイル中のデータの中に、**マークが現れることがありますが、これはその値がまだ確定していないことを示しています。具体的にはアブソリュート値で処理されていないラベルを用いたときに現れます。

かと思ったのですが、近いうちにZ80's Barのほうで詳しくやるようなので、そちらを見てください。

また、WZD1.ASMは、最初Small-Cで書いておいてその後、私がASMファイルを書き直したものです。ラベルにCCxxというのが多く使われているのは、そのためです。この方法は、いきなりマシン語のコーディングをしなくてすむし、かといってフローチャートのようなものを細かく書かなくていいのでたいへん便利です。

ただニーモニックからマシン語への変換をくそ真面目に行うプログラムなので、特にテクニックのようなものはそんなにありません。ひたすら、腕力の勝負でした。

このようなプログラムを作るコツは、それぞれのルーチンができるだけ一般化してしまうということです。

たとえば、パラメータは、HLレジスタ、DEレジスタ、BCレジスタの順に用いて戻り値は、HLレジスタに代入して返すとか、原則的に各ルーチンは、戻り値を代入するレジスタ以外は保存しておくとか、決めておくといろいろと便利です。それから、ポインタはなるべくあと戻りさせないというのもわかりやすいプログラムを書くうえでけっこう有効です。

このプログラムを作っていて気がついたものとしては、すべてのレジスタを破壊しないで、ある（メモリ上に取ってある）スイッチの値を調べるときには、

```
PUSH    HL
LD       HL, スイッチのアドレス
LD       H, (HL)
INC      H
DEC      H
POP      HL
```

；ここでスイッチの値について（0かそれ以外か）Zフラグが変化すると方法があるということです。1パス目か2パス目かの判断に多用しています。

ファイル関係のサブルーチンを今回は自作したのですが、なるべくCP/MのFCB(File Control Blockの略だっけ)に似せようと思ったのであまり効率のよいものではありません。が、このようなサブルーチンを作成するとき、2カ所以上で無関係に同じファイルに書き込みを行おうとした場合の処理を、見落としがちなので少しばかり触れておきます（以前発表されたSLANG用ファイル入出力サブルーチンもアプリケーションのほうで処理してやらなくてはならないようです）。

まず、システムが使用していないファイ

ル属性からひとつの値を選びそれを「書き込み中」を表す属性にしてやります（今回は5にしました）。

ファイルを書き込み用にオープンしたときは、ファイルに対してその属性を与えてやります。こうしてやれば、次の書き込み用ファイルをオープンする前にそのファイルの属性は「書き込み中」を表していないかどうか、調べるだけですみます。特にS-OSの場合は、わざわざ調べなくてもファイルのオープン時に、Bad File Modeエラーが発生します。

そして、クローズするときに元々の属性に戻してやれば、(ASCIIファイルなら4とか、バイナリファイルであれば1とか)ちゃんとディスクに収まるのです。

WZD をソースリストからアセンブルするときは、

```
#LWZD
*=WZD1
*=WZD2
*=WZD3
*=WZD35
*=WZD4
*[ここで、シフト+ブレイクを押す]
以上により、WZD1.REL, WZD2.REL, WZD3.REL, WZD4.RELが、作成されますので、次にこれらのファイルを、
# WLK
*/P:3000,/D:6000
*WLK1,WLK2,WLK3,WLK4,WLK/N:P
```

としてリンクすれば、ここに掲載されているリストと同じものがWZD.OBJというファイルに得られます。

プログラムにはアルゴリズムを表している部分とデータを表している部分がありますが（特にマシン語では露骨にそうなる）、ときとして両者を分けたいときがあります。アブソリュートアセンブラならワークエリ

アを後ろに持っていくとかすればよいのですが、複数のリロケータブルファイルからオブジェクトファイルを作る場合はそうはいきません。そこでリンクに複数のPCを持たせてアルゴリズムを表すコードがきたら、コードセグメントを受け持つPCを用いて、データの部分はデータセグメントを受け持つPCを用いて処理してやります。

なお、WZDでは、コードセグメントとデータセグメントのほかにワークセグメントが、用意されています。MACRO-80で用意されていたアブソリュートセグメントは、用意されていません。具体例はWZD自身のソースプログラムを見てください。

最後に

この記事の内容がよくわからなかったとしても、それは気にするほどのことではないと思います。WZDを使っているうちに、なんとなくわかってくるから、きっと。なお質問点があれば質問箱にお寄せください（初心者・女性大歓迎、ひやかし・オタク小歓迎）。ひよっとすると、来々月号でWZDに関する質問特集をやるかもしれません。

それから、私に逆ポーランドから普通の数式に変換するアルゴリズムを教えてください。宝塚市のTAOさん、熊本市の茶円亮さん、PurePASCALの藤本健士さん、TED-750の鈴木典雄さんありがとうございました。

「材料の力学」という科目の単位を修得しだい、逆アセンブラの作成に取り掛かろうと思います。でも、なんかコンパイラをその前にひとつ作りたいな、んん、その前に単位が先かなあ？

では、来月のWLKとWLBの発表でお会いしましょう（それまで、16ビット機に乗り移っちゃ駄目だよ！）。

セグメントについて

セグメントの異なるエリアに、JR命令やDJNZ命令でジャンプすることはできません（JR命令やDJNZ命令のオペランドの値は、WZD内で処理を完了する必要がある。その処理をリンクに持ち越してはいけません）。

コードセグメントとデータセグメントは、アドレスを、重ねてはいけませんが、ワークセグメントは、コードセグメントおよびデータセグメントと重ねてかまいません。

また、ワークセグメントの内容は、リンクにアドレス情報を伝えるだけで、実体は作成することができません。さらに例外的にこのセグメントだけPCのあと戻りができます。

いったんORG命令を処理すると同一ファイル内においてはそのセグメントのアドレス情報は、

すべてアブソリュート値で処理されます（MACRO-80でいうところの、ASEGです）。そのようなときには、前述の理由により、ORG命令を挟んで相対ジャンプはできません（絶対ジャンプはできます）。

S-OSでは、64Kバイトを超える大きさのファイルを扱うことはできません。ソースファイルの大きさが64Kバイトを超えることはまずないと思いますが（だってエディタが扱えないでしょ）、30Kバイトくらいのソースファイルからリストファイルを作った場合、64Kバイトを超えてしまう可能性がありますので注意してください。そのときはWZDは、“Bad Record”と、エラーメッセージを表示してすべてのアセンブル作業を中止します。

表1 リファレンスマニュアル

パラメータ

基本的には、

[リロケータブルファイル名], [リストファイル名] =
[ソースファイル名]

の書式でパラメータを与えるが、続けて以下のスイッチを併用することによってパラメータの一部またはソースファイルの一部を省略することができる。

なお、[リストファイル名], [リロケータブルファイル名], [ソースファイル名] において拡張子名を省略した場合にはそれぞれ、'.PRN', '.REL', '.ASM' になります。

/L

リストファイルのファイル名をソースファイルのファイル名と同じにする。

/R

リロケータブルファイルのファイル名をソースファイルのファイル名と同じにする。

/K

ソースファイルには、シフトJISコードで表される漢字などが含まれていることを示す。具体的には、以下のようになります。

= [ファイル名]

ソースファイルとリロケータブルファイルを [ファイル名] にする。

, = [ファイル名]

ソースファイルを [ファイル名] にする。このときリロケータブルファイルとリストファイルは作らない。ソースファイル中のアセンブルエラーとなる箇所をチェックするのに便利。

= [ファイル名] /L

ソースファイル, リロケータブルファイル, リストファイルすべてを [ファイル名] とする。

, = [ファイル名] /L

ソースファイルとリストファイルを [ファイル名] にする。

[ファイル名1] = [ファイル名2]

ソースファイルを [ファイル名2] に、リロケータブルファイルを [ファイル名1] にする。

[ファイル名1] = [ファイル名2] /L

ソースファイル, リストファイルを [ファイル名2] に、リロケータブルファイルを [ファイル名1] にする。

, [ファイル名1] = [ファイル名2]

ソースファイルを [ファイル名2] に、リストファイルを [ファイル名2] にする。

, [ファイル名1] = [ファイル名2] /R

ソースファイル, リロケータブルファイルを [ファイル名2] に、リストファイルを [ファイル名1] にする。

疑似命令

REDA に比べて以下の命令が拡張されています。

END [式]

アセンブル作業を終了します。この行の後ろになにを書いてもアセンブラには無視されます。式がある場合にはその値を実行開始アドレスに指定します ([式] は省略可能)。

.KANJI

以後のソースプログラムはシフトJISコードで書かれていることを示します (パラメータのスイッチで/Kを指定した場合と同じ) 漢字を用いるときには、先立ってこの命令も書いておいてください。

.NKANJI

以後のソースプログラムはASCIIコードのみで書かれていることを示します。先の.KANJI命令と組み合わせることによって、REDAにおけるKスイッチの働きをアセンブル中に切り替えられます。

CSEG

以下のプログラムはコードセグメントに配置されることを示します。

DSEG

以下のプログラムはデータセグメントに配置されることを示します。

WSEG

以下のプログラムはワークセグメントに配置されることを示します。

EXT [ラベル名]

[ラベル名] で表されたラベルはこのファイルにはなく、他のファイルで外部ラベルとして宣言されていることを示します。同じファイル中に同じ名前があった場合は、Multi Defined エラーとなります。

PUBLIC [ラベル名]

[ラベル名] で表されたラベルを外部 (参照可能) ラベルとして宣言します。同じファイル中に [ラベル名] で表されたラベルがない場合は、Undefined Label エラーとなります。

[ラベル名] #

[ラベル名] で表されたラベルは、外部ラベルとして処理されます。なお、この表記は式中においてのみ可能です。

[ラベル名] :

[ラベル名] で表されたラベルを、現時点でのPCの値を表す内部ラベルとして宣言するとともに、外部 (参照可能) ラベルとしても宣言します。この表記は、ソースプログラム中の1列目においてのみ可能です。

.PHASE [式]

以下のプログラムのPCを一時的に [式] の値にします。ただ、PCを一時的に変更するだけで、オブジェクトプログラムは以前のアドレスと連続して配置されます (プログラムの転送は行われませんので、プログラマの責任において行ってください)。

.DEPHASE

PHASE文で一時的に変更したPCを元の値に戻します (元の値といってもPHASE文とDEPHASE文のあいだのプログラム分だけPCは進んでいます)。

INCLUDE [ファイル名]

ソースプログラム中のこの位置に [ファイル名] で示されたファイルをそっくり読み込みます。読み込まれたファイルの中でさらにINCLUDE文が使われていても (つまりINCLUDE文がネ스팅していても) 4重までなら構いません。

PAGE [式]

リストファイルにおいて1ページあたりの行数を指定します。ページごとに区切られたくない場合は、PAGE OFFFHとしておいてください。現実的には区切られません (6万行以上からなるファイルは存在しない、と思う)。デフォルトの [式] の値はOFFFHです。

TITLE [文字列]

リストファイルにおいてこのプログラムのタイトルを [文字列] にします。ここで、タイトルとはリストファイルの一番最初の行にプリントされるタイトルであって、リストファイルのファイルネームとは関係ありません。

表2 エラーメッセージ

Syntax Error

文法エラーが発生した。

Undefined Label Error

未定義なラベルが使用された。

Redefinition Error

同じラベルを2度以上定義しようとした。

Illegal Label Error

ラベル名がおかしい。

Illegal Opcode Error

オペコードがおかしい。

Illegal Operand Error

オペランドがおかしい。

Too Many Labels Error

使用されたラベルが多すぎる (たいていの

機種では、まず起こらない)。

Missig Label Error

ラベルのあるべきところにラベルがない。

Too Far Error

JRやDJNZ命令などでオペランドで表された値が届く範囲にない。

Illegal Expression Error

式の表し方がおかしい。

Missing [] Error

" " の文字がない。

Missing [.] Error

". " の文字がない。

Missing Quote Error

文字列の表し方がおかしい。

Illegal ORG Error

ORG命令のオペランドがおかしい (アドレスの低いほうに指定しなさいなど)。

Value Error

1パス目で決まっていなければならない値が決まっていない (EQU, ORG命令など)。

Relocation Error

JRまたはDJNZ命令のオペランドにおいて、値が不確定な意味のない式を書いた (JR Label * 2 など)。

Internal Error

WZDの内部で起きたエラー (このエラーがもし発生したらそれは、WZD内にバグが潜んでいるということです。そのときは、Oh! X編集部経由で私に知らせてください)。

リスト1 WZDダンブルリスト

3000 ED 73 00 60 ED 7B 6A 1F : B1
 3008 ED 5B 76 1F 06 01 1A 13 : 11
 3010 A7 28 07 FE 20 20 F7 04 : 0F
 3018 18 F4 78 32 02 60 AF 32 : F9
 3020 57 60 32 9F 60 32 B1 60 : 2B
 3028 32 A3 61 32 DB 61 3A 02 : E0
 3030 60 FE 01 20 05 CD 6C 31 : EE
 3038 18 0C 2A 76 1F 7E 23 FE : 82
 3040 20 20 FA 22 03 60 21 88 : 68
 3048 31 CD E1 33 11 57 60 2A : 04
 3050 03 60 CC 13 34 11 B1 60 : 98
 3058 2A 03 60 CD 13 34 22 03 : C6
 3060 60 21 8A 31 CD E1 33 11 : 2E
 3068 9F 60 2A 03 60 CC 13 34 : 9F
 3070 22 03 60 21 88 31 CD E1 : 0D
 3078 33 11 57 60 2A 03 60 CC : 54

SUM: 6C DC 25 00 AE B7 6B 00 ECCF

3080 13 34 22 03 60 2A 03 60 : 59
 3088 7E A7 CA D1 30 21 8C 31 : CE
 3090 CD E1 33 28 08 21 8F 31 : F2
 3098 CD E1 33 20 0C 11 B1 60 : 2F
 30A0 21 57 60 CD 13 34 C3 85 : 34
 30A8 30 21 92 31 CD E1 33 28 : 1D
 30B0 08 21 95 31 CD E1 33 20 : F0
 30B8 0C 11 9F 60 21 57 60 CD : C1
 30C0 13 34 C3 85 30 21 A4 31 : B5
 30C8 CD 4F 34 CD 5A 34 C3 85 : F3
 30D0 30 3A 57 60 A7 11 57 60 : 90
 30D8 21 98 31 C4 F5 33 3A B1 : C1
 30E0 60 A7 11 B1 60 21 9C 31 : 17
 30E8 C4 F5 33 3A 9F 60 A7 11 : DD
 30F0 9F 60 21 A0 31 C4 F5 33 : DD
 30F8 11 57 60 21 9F 60 CD 26 : DB

SUM: 95 EF BC CD 67 08 55 1E 4E99

3100 34 21 C0 31 D4 38 34 11 : 97
 3108 57 60 21 B1 60 CD 26 34 : 10
 3110 21 C0 31 D4 38 34 11 9F : 02
 3118 60 21 B1 60 CD 26 34 21 : DA
 3120 C0 31 D4 38 34 3A 57 60 : 22
 3128 A7 21 C0 31 CC 38 34 CD : BE
 3130 02 32 CD 47 32 2A 3D 62 : 43
 3138 7C B5 F5 C4 19 48 F1 21 : 5D
 3140 58 31 CC 4F 34 21 5B 31 : 85
 3148 CD 4F 34 3A 02 60 FE 01 : EB
 3150 CA 1E 30 ED 7B 00 60 C9 : A9
 3158 4E 6F 00 20 46 61 74 61 : 59
 3160 6C 20 45 72 72 6F 72 28 : BE
 3168 73 29 0D 00 3E 2A CD F4 : D2
 3170 1F ED 5B 76 1F CD D3 1F : BB
 3178 1A FE 1B CA 5A 34 13 1A : B8

SUM: 46 DC 11 D2 A4 BF AA 66 DFA2

3180 A7 28 E9 ED 53 03 60 C9 : 24
 3188 3D 00 2C 00 2F 72 00 2F : 39
 3190 52 00 2F 6C 00 2F 4C 00 : 68
 3198 41 53 4D 00 52 45 4C 00 : C4
 31A0 50 52 4E 00 55 73 61 57 : 80
 31A8 65 3A 20 57 5A 44 20 5B : 2F
 31B0 52 45 4C 5D 2C 5B 50 52 : 69
 31B8 4E 5D 3D 41 53 4D 0D 00 : D6
 31C0 53 6F 75 72 63 65 20 66 : F7
 31C8 69 6C 65 20 6E 61 6D 65 : FB
 31D0 20 69 73 20 6E 6F 74 20 : 8D
 31D8 67 69 76 65 6E 0D 00 41 : 67
 31E0 53 4D 09 66 69 6C 65 20 : 69
 31E8 3A 00 0D 52 45 4C 20 66 : B0
 31F0 69 6C 65 20 3A 00 0D 50 : F1
 31F8 52 4E 20 66 69 6C 65 20 : 80

SUM: 57 5D E6 A3 00 AE CE 2E 3964

3200 3A 00 AF 32 2E 62 32 4F : 2C
 3208 62 CD EC 44 CD 4B 44 21 : DC
 3210 00 00 22 55 60 3E 04 11 : 2A
 3218 57 60 21 C3 60 CD D7 4C : EB
 3220 21 57 60 DA 71 33 3E FF : 93
 3228 32 D9 4E CD 90 33 3E FF : 2C
 3230 3A 4F 62 A7 20 05 CD E2 : 66
 3238 37 18 F0 3A 55 60 3D 32 : 9D
 3240 55 60 3C C2 2B 32 C9 3E : 17
 3248 01 32 2E 62 AF 32 4F 62 : 55
 3250 32 55 60 CD 67 44 3E 04 : A1
 3258 11 57 60 21 C3 60 CD D7 : B0
 3260 4C 21 57 60 DC 71 33 3E : E2
 3268 FF 32 D9 4E 3A B1 60 A7 : 4A
 3270 28 15 3E 01 4D B1 60 21 : BF
 3278 DB 61 CD C0 11 21 B1 60 : 48

SUM: 9E CB 43 97 A9 7F 98 CC E129

3280 DA 71 33 AF 32 48 4F 3A : 30
 3288 9F 60 32 3F 62 A7 28 15 : B6
 3290 3E 04 11 9F 60 21 A3 61 : 77
 3298 CD C0 4D 21 9F 60 DA 71 : 45
 32A0 33 AF 32 74 4F CD 90 33 : 67
 32A8 38 0B 3A 4F 62 A7 20 05 : FA
 32B0 CD E2 37 18 F0 3A 55 60 : DD
 32B8 3D 32 55 60 C2 A5 32 : F9
 32C0 3A B1 60 A7 28 14 CD 84 : 7F
 32C8 35 3E FF CD B5 33 21 00 : 48
 32D0 BC 22 25 50 21 DB 61 CD : 7D
 32D8 3B 4E 3A 9F 60 A7 C8 3E : 6F
 32E0 00 CD C8 33 21 00 BD 22 : C8

32E8 25 50 21 A3 61 CD 3B 4E : F0
 32F0 C9 ED 53 6D 33 3A 55 60 : 98
 32F8 3C 32 55 60 FE 04 38 09 : 66

SUM: 89 FE 0A EF 81 B4 3A 53 AABA

3300 21 59 33 CD 4F 34 C3 5A : 1A
 3308 34 2A 55 60 01 12 00 CD : F3
 3310 46 48 11 57 60 19 22 6F : 00
 3318 33 ED 5B 6D 33 1A CD 3F : 41
 3320 34 38 05 13 77 23 18 F5 : 2B
 3328 36 00 ED 53 6D 33 01 38 : 4F
 3330 00 2A 55 60 CD 46 48 11 : 4B
 3338 C3 60 19 3E 04 ED 5B 6F : 35
 3340 33 CD D7 4C 2A 6F 33 DA : C9
 3348 71 33 2A 55 60 29 11 D9 : 96
 3350 4E 19 36 FF ED 5B 6D 33 : 84
 3358 C9 49 4E 43 4C 55 44 45 : CD
 3360 20 6C 65 76 65 6C 20 6F : C7
 3368 76 65 72 20 0D 00 00 00 : 7A
 3370 00 E5 21 82 33 CD 4F 34 : 0B
 3378 E1 CD 4F 34 CD EE 1F C3 : CE

SUM: 2D 5F 20 24 CD 71 F1 13 F59D

3380 5A 34 43 61 6E 20 6E 6F : 9D
 3388 74 20 6F 70 65 6E 20 00 : 66
 3390 06 50 21 05 60 C5 E5 CD : 53
 3398 AA 4E E1 C1 A7 28 0B FE : 72
 33A0 0D 28 0B 77 23 05 20 ED : EC
 33A8 B7 C9 AF 77 37 C9 3E 0D : F1
 33B0 77 23 AF 77 C9 E5 21 B1 : 40
 33B8 60 6E 2C 2D E1 C8 CD 1E : BB
 33C0 4F D0 CD 33 20 C3 5A 34 : 90
 33C8 E5 B7 21 3F 62 6E 2C 2D : 25
 33D0 28 07 D5 C5 CD 4A 4F C1 : F0
 33D8 D1 E1 D0 CD 33 20 C3 5A : BF
 33E0 34 ED 5B 03 60 7E A7 28 : 2C
 33E8 07 1A BE C0 13 23 18 F5 : E2
 33F0 ED 53 03 60 C9 06 0D 1A : 99
 33F8 CD 3F 34 38 0A 05 28 07 : B6

SUM: 3B 7C 2C 88 A6 3D 56 BD 7A89

3400 FE 2E 28 03 13 18 F0 1A : 8C
 3408 FE 2E C8 3E 2E 12 13 CD : 52
 3410 13 34 C9 06 0D AF 12 7E : 62
 3418 CD 3F 34 D8 05 CD 87 23 : 86
 3420 12 13 AF 12 18 F1 1A A7 : B0
 3428 37 C8 7E A7 37 C8 1A A7 : E4
 3430 C8 13 BE 23 37 C0 18 F6 : C1
 3438 CD 4F 34 CD 5A 34 C9 FE : 72
 3440 3D 28 0A FE 2F 28 06 FE : C8
 3448 2C 28 02 A7 C0 37 C9 7E : 3B
 3450 23 A7 C8 E5 CD F4 1F E1 : 38
 3458 18 F5 3A DB 61 A7 21 00 : 4B
 3460 BC 22 25 50 21 DB 61 C4 : 74
 3468 3B 4E 3A A3 61 A7 21 00 : 8F
 3470 BD 22 25 50 21 A3 61 C4 : 3D
 3478 3B 4E ED 7B 00 60 C9 CD : E7

SUM: 4D D8 8B EB F3 CD 63 7C 97D1

3480 97 44 E5 21 2E 62 6E 2D : 0C
 3488 E1 C0 F5 3A 9F 60 A7 28 : 9E
 3490 08 F1 F5 CD 28 37 CD 62 : 49
 3498 37 3A 52 62 FE 60 20 20 : C3
 34A0 F1 D5 E5 ED 5B 50 62 16 : BB
 34A8 00 21 53 62 19 77 7B E1 : C2
 34B0 D1 3C E6 1F 32 62 C0 62 : B6
 34B8 C5 06 20 CD 6A 35 C1 C9 : E1
 34C0 C5 3A 50 62 47 3A 52 62 : E6
 34C8 A7 C4 6A 35 C1 3E 60 32 : 9B
 34D0 52 62 3E 01 32 50 62 F1 : C8
 34D8 32 53 62 C9 C5 06 01 FE : 7A
 34E0 84 28 10 FE EA 28 0C 06 : D8
 34E8 02 FE 82 28 06 FE 55 28 : FD
 34F0 02 18 0E F5 3A 9F 60 A7 : BB
 34F8 C4 6A 37 CD 97 44 10 FB : 18

SUM: 7A C2 90 0E BD 7C 78 AA F33F

3500 F1 C1 E5 21 2E 62 6E 2D : E3
 3508 E1 C0 F5 C5 CD 84 35 C1 : A2
 3510 F1 CD 90 35 C9 CD DC 3A : 29
 3518 F5 7D CD 02 35 7C CD 02 : C1
 3520 35 F1 C9 CD 97 44 E5 21 : 9D
 3528 2E 62 6E 2D E1 C0 E5 21 : D2
 3530 52 62 BE E1 20 1A D5 E5 : 47
 3538 3A 50 62 16 00 5F 21 53 : D5
 3540 62 19 71 E1 D1 3C E6 07 : C7
 3548 32 50 62 C0 06 08 18 1A : E4
 3550 F5 3A 50 62 47 3A 52 62 : 16
 3558 A7 C4 6A 35 F1 32 52 62 : E1
 3560 3E 01 32 50 62 79 32 53 : 21
 3568 62 C9 04 05 C8 3A 52 62 : EA
 3570 80 3D CD 90 35 D5 E5 21 : 2A
 3578 53 62 7E CD 90 35 23 10 : F8

SUM: 4A A0 9C F8 8F 19 3A 69 CCB8

3580 F9 E1 D1 C9 3A 50 62 47 : A7
 3588 CD 6A 35 AF 32 50 62 C9 : C8
 3590 E5 D5 C5 F5 CD B5 33 F1 : 1A
 3598 C1 D1 E1 C9 E5 21 2E 62 : D2
 35A0 6E 2D E1 C0 D5 E5 F5 3A : 25
 35A8 51 62 6F 3C 32 51 62 26 : 69
 35B0 00 11 73 62 19 F1 77 E1 : 48

35B8 D1 C9 CD 9C 35 F5 7D CD : 77
 35C0 9C 35 7C CD 9C 35 F1 C9 : A5
 35C8 3A 51 62 21 73 62 A7 C8 : 52
 35D0 F5 7E 23 CD 02 35 F1 3D : C8
 35D8 18 F4 3A A5 62 E6 01 20 : 54
 35E0 09 79 CD 7F 34 78 CD 7F : C6
 35E8 34 C9 CD C8 35 3E 82 CD : 54
 35F0 DC 34 C9 3A A5 62 E6 01 : 01
 35F8 79 CA 7F 34 CD C8 35 3E : FE

SUM: 71 92 59 45 C1 24 64 EA BA5D

3600 84 CD DC 34 C9 3A 2E 62 : F4
 3608 A7 C8 3A 9F 60 A7 C8 AF : C6
 3610 32 13 62 3E 01 32 14 62 : 8E
 3618 C3 79 36 3A 2E 62 A7 C8 : AB
 3620 3A 9F 60 A7 C8 AF 32 13 : 9C
 3628 62 CD 62 37 CD 3F 37 CD : D8
 3630 62 37 CD 62 37 C9 3A 2E : 30
 3638 62 A7 C8 3A 9F 60 A7 C8 : 79
 3640 E5 21 20 20 22 16 62 22 : 02
 3648 18 62 22 1A 62 3E 07 32 : 8F
 3650 13 62 E1 C9 F5 C5 D5 E5 : 93
 3658 11 15 62 2A 13 62 26 00 : 4D
 3660 19 77 23 36 00 3A 13 62 : 98
 3668 3C 32 13 62 FE 16 D4 76 : 41
 3670 36 E1 D1 C1 F1 C9 CD 89 : B9
 3678 36 CD 62 37 CD 5C 3A CD : CC

SUM: 62 BC F3 82 0B 7C 4D 78 2F62

3680 3F 37 CD 4A 37 CD 62 37 : 2A
 3688 C9 3A 2E 62 A7 C8 3A 9F : DB
 3690 60 A7 C8 2A A8 62 7C B5 : 34
 3698 CC DC 36 2A A8 62 23 22 : 57
 36A0 A8 62 2B D5 ED 5B A6 62 : 5A
 36A8 B7 ED 52 D1 38 06 21 00 : 26
 36B0 00 22 A8 62 CD 00 37 21 : 51
 36B8 15 62 7E A7 28 06 23 CD : BA
 36C0 C8 33 18 F6 3A 14 62 A7 : 60
 36C8 28 05 CD 86 37 18 05 35 : 12
 36D0 0D CD C8 33 AF 32 13 62 : 2B
 36D8 32 14 62 C9 21 00 00 22 : B4
 36E0 A8 62 3A AA 62 A7 C8 21 : E0
 36E8 AA 62 7E 23 A7 28 05 CD : 4E
 36F0 C8 33 18 F6 30 0D CD C8 : E9
 36F8 33 21 01 00 22 A8 62 C9 : 4A

SUM: 24 F8 7C EA F2 A2 D2 E5 9E5D

3700 3A FB 62 A7 C8 21 17 37 : 75
 3708 D5 16 00 5F 19 7E D1 32 : E4
 3710 15 62 AF 32 FB 62 C9 53 : D1
 3718 55 4D 53 53 41 53 45 41 : 62
 3720 53 53 53 49 41 56 52 49 : 74
 3728 F5 0F 0F 0F 0F CD 31 37 : 66
 3730 F1 E6 0F C6 30 FE 3A 38 : 4C
 3738 02 C6 07 CD 54 36 C9 F5 : E4
 3740 7C CD 28 37 7D CD 28 37 : 51
 3748 F1 C9 C5 3A 45 62 06 27 : 8D
 3750 FE 02 28 08 06 22 FE 03 : 59
 3758 28 02 06 21 78 CD 54 36 : 20
 3760 C1 C9 F5 3E 20 CD 54 36 : 34
 3768 F1 C9 E5 21 2E 62 6E 2D : EB
 3770 E1 C0 F5 C5 3E 2A CD 54 : EA
 3778 36 3E 2A CD 54 36 CD 62 : 24

SUM: 10 F8 F0 01 11 58 58 5A A2D4

3780 37 10 F1 C1 F1 C9 3A 13 : 00
 3788 62 FE 08 CD 37 FE 10 10 : 59
 3790 DC D0 37 3A 9F 60 A7 28 : EB
 3798 23 3A 55 60 A7 28 1D 3A : 38
 37A0 13 62 47 3E 16 90 47 C5 : AC
 37A8 3E 20 CD C8 33 C1 10 7F : EE
 37B0 3E 43 CD C8 33 3E 20 CD : 74
 37B8 C8 33 18 08 3A 13 62 FE : C8
 37C0 18 DC D0 37 11 05 60 1A : 8B
 37C8 13 A7 C8 CD C8 33 18 F7 : 59
 37D0 F5 3E 09 CD C8 33 3A 13 : 51
 37D8 62 C6 08 E6 F8 32 13 62 : B5
 37E0 F1 C9 ED 73 2F 62 3A 9F : 84
 37E8 60 A7 C4 05 36 CD 5C 3A : 69
 37F0 22 39 62 2A 33 62 23 22 : C1
 37F8 33 62 11 05 60 ED 53 31 : 7C

SUM: 17 A2 4B 6B 4E 45 A6 BE F689

3800 62 CD 5D 38 CD 53 43 1A : 41
 3808 FE 3B 28 3D FE 0D 28 39 : 0A
 3810 FE 2E 28 44 D6 41 DA 95 : 1E
 3818 48 FE 1A D2 95 48 87 4F : R5
 3820 06 00 21 5E 4A 09 7E 23 : 79
 3828 66 6F EB CD 77 3A EB DA : 03
 3830 95 48 CD 53 43 08 CD 9E : B3
 3838 3A CD 53 43 1A A7 28 0F : 95
 3840 FE 0D 28 0B FE 3B C2 8A : C3
 3848 48 1A 13 FE 0D 20 FA 3A : D4
 3850 9F 60 A7 C4 89 36 B7 C9 : A9
 3858 21 B5 4B 18 CD 1A FE 40 : 5E
 3860 D8 CD 2E 3A CD 53 43 21 : 91
 3868 81 38 CD 1F 3A 28 19 21 : 41
 3870 84 38 CD 1F 3A 28 63 1A : 87
 3878 13 FE 3A CA 0D 39 C3 8A : A8

SUM: D7 2F 22 73 03 62 1D 94 CE2B

3880 48 3A 3A 00 45 51 55 00 : A7

4178 3A CD 12 3A 78 FE 20 D2 : BB
SUM: 1D F5 97 3A 45 6C 8E 0D D44D

4180 92 48 F6 20 CD DC 34 1A : E7
4188 CD 36 3A 38 06 13 CD DC : 37
4190 34 18 F4 3E 81 CD 15 35 : 16
4198 2A 0E 3A 23 2F AF CD 9A : CE
41A0 1F C9 3A 2E 62 A7 CA 49 : 6C
41A8 38 CD 53 43 D5 CD B8 39 : 2E
41B0 DA 8C 48 D1 FE 01 20 13 : B1
41B8 60 69 3E E7 CD 15 35 2A : 2F
41C0 3B 62 23 22 3B 62 3E 81 : 3E
41C8 CD 15 35 CD 12 3A 78 FE : A6
41D0 20 D2 92 48 CD 15 35 1A : FD
41D8 CD 36 3A 38 06 13 CD DC : 37
41E0 34 18 F4 CD 5D 43 28 C1 : 96
41E8 C9 3E 01 32 4F 62 CD 42 : FA
41F0 44 C8 CD FC 4A CD C8 35 : E3
41F8 3E E9 CD DC 34 C9 3A 41 : 48
SUM: C2 B5 24 28 BD F4 69 72 68D2

4200 62 E6 02 C2 B9 48 3E 02 : 4D
4208 32 45 62 3A 42 62 32 41 : 2A
4210 62 2A 46 62 22 4C 62 3E : 42
4218 E1 CD DC 34 3E 00 CD DC : A5
4220 34 C9 3A 41 62 E6 02 C2 : 84
4228 B9 48 3E 03 32 45 62 3A : 55
4230 43 62 32 41 62 2A 48 62 : 4E
4238 22 4C 62 3E E1 CD DC 34 : CC
4240 3E 01 CD DC 3A C9 3A 41 : 60
4248 62 E6 02 C2 B9 48 3E 04 : 4F
4250 32 45 62 3A 42 62 32 41 : 2C
4258 62 2A 4A 62 22 4C 62 3E : 46
4260 E1 CD DC 34 3E 02 CD DC : A7
4268 34 C9 CD 53 43 1A CD DC : C9
4270 43 20 05 CD B0 42 18 06 : 45
4278 CD FC 44 CD F3 35 CD 5D : 2C
SUM: 82 E9 FF B0 A9 6A B2 74 3241

4280 43 28 E7 C9 CD 53 43 CD : 4B
4288 FC 44 3A A5 62 E6 01 20 : 88
4290 0A 79 CD 7F 34 78 CD 7F : C7
4298 34 18 08 CD C8 35 3E E5 : 41
42A0 CD DC 34 CD 5D 43 28 DC : 4E
42A8 C9 1A CD 82 43 C2 AD 48 : 2C
42B0 3A 46 62 A7 20 28 13 1A : F8
42B8 FE 0D CA AD 48 FE 5E 28 : 4E
42C0 18 CD 82 43 20 0E 13 4F : 3A
42C8 1A CD 82 43 28 04 CD 6A : 0F
42D0 43 C8 1B 79 CD 7F 34 18 : 37
42D8 DD CD 1A 43 18 F6 13 1A : 42
42E0 FE 0D CA AD 48 FE 5E 28 : 4E
42E8 2C CD 44 43 38 0F CD 7F : 13
42F0 34 13 1A FE 0D CA AD 48 : 2B
42F8 CD 7F 34 18 E1 CD 82 43 : 0B
SUM: C8 DB B8 A5 CE 3C 16 D4 088B

4300 20 0E 13 4F 1A CD 82 43 : 3C
4308 28 04 CD 6A 43 C8 1B 79 : 02
4310 CD 7F 34 18 C9 CD 1A 43 : 8B
4318 18 F6 13 1A FE 0D CA AD : BD
4320 48 D6 40 38 03 FE 1C D8 : 8B
4328 C6 40 C9 CD FC 4A CD C8 : 71
4330 35 3E B0 CD DC 34 3A 45 : DF
4338 62 E6 01 C2 B6 48 60 69 : D2
4340 CD BC 44 C9 FE 81 D8 FE : EB
4348 A0 3F D0 FE E0 D8 FE F0 : 53
4350 3F C9 13 1A FE 09 28 FA : 5E
4358 FE 20 28 F6 C9 CD 53 43 : 68
4360 FE 2C 28 03 FE 3A C0 13 : 60
4368 C9 1A D9 21 C9 4C 01 0E : 01
4370 00 ED B1 D9 C9 1A FE 09 : 61
4378 C8 FE 0D C8 FE 3B C8 FE : 9A
SUM: 0B D6 EF 1B E8 37 DC AD E462

4380 20 C9 FE 27 C8 FE 22 C9 : BF
4388 21 42 4C 06 08 18 13 CD : B5
4390 96 43 D0 C3 98 48 21 11 : 7E
4398 4C 06 11 18 05 21 20 4C : 0D
43A0 06 08 C5 48 D5 1A BE 20 : E8
43A8 04 13 23 18 F8 34 35 20 : D3
43B0 0A CD 6A 43 20 05 E1 79 : 03
43B8 90 C1 C9 CD F8 43 D1 10 : 03
43C0 E3 C1 37 C9 34 35 37 C8 : 0C
43C8 4B 42 1A BE 20 05 13 23 : C0
43D0 C3 CA 43 34 35 23 20 05 : 81
43D8 CD 76 43 28 0A CD F8 43 : C0
43E0 23 23 59 50 C3 CA 43 4E : 07
43E8 23 46 78 B7 20 05 CD D3 : 5D
43F0 3A B7 C9 CD CB 3A B7 C9 : 0C
43F8 AF BE 23 C8 C3 F9 43 CD : 24
SUM: B4 1E DA F7 56 3B 87 A6 AA6F

4400 1F 4A AF 91 4F C9 CD D7 : 5F
4408 3A 1A 13 FE 2B 28 10 FE : C6
4410 2D 28 EC FE 29 C2 98 48 : 0A
4418 AF 32 A5 62 0E 00 C9 CD : 8C
4420 FC 44 1A FE 29 C2 A7 48 : 32
4428 13 C9 CD 3C 4A C2 98 48 : CB
4430 18 ED 47 1A FE 2C 78 C2 : CA
4438 AA 48 13 C9 1A FE 28 C0 : CE
4440 13 C9 CD 53 43 FE 0D C8 : 12
4448 FE 3B C9 21 00 00 22 3B : 80
4450 62 22 A8 62 22 3D 62 21 : 70
4458 00 02 22 35 62 21 FF FF : DA
4460 22 A6 62 AF 32 AA 62 AF : C6

4468 32 50 62 32 52 62 32 42 : 3E
4470 62 32 43 62 32 44 62 32 : 43
4478 41 62 3E 02 32 45 62 3E : FA
SUM: 70 AC 39 5C E5 52 05 80 EBC4

4480 00 32 40 62 21 00 00 22 : 17
4488 33 62 22 46 62 22 48 62 : 2B
4490 22 4A 62 22 4C 62 C9 F5 : 5C
4498 E5 3A 45 62 21 46 62 FE : 8D
44A0 02 28 0A 21 48 62 FE 03 : 00
44A8 28 03 21 4A 62 34 20 02 : 4E
44B0 23 34 2A 4C 62 23 22 4C : C0
44B8 62 E1 F1 C9 E5 ED 4B 4C : 66
44C0 62 09 22 4C 62 E1 3A 45 : 9B
44C8 62 FE 02 20 09 ED 4B 46 : 09
44D0 62 09 22 46 62 C9 FE 03 : FF
44D8 20 09 ED 4B 48 62 09 22 : 36
44E0 48 62 C9 ED 4B 4A 62 09 : 60
44E8 22 4A 62 C9 21 00 00 01 : B9
44F0 01 02 AF CD 9A 1F ED A1 : C6
44F8 EA F3 44 C9 AF 32 51 62 : 7E
SUM: 84 12 A0 F5 AB 04 2A D1 7D41

4500 32 A5 62 CD 2B 45 CD 53 : 96
4508 43 FE 2B 20 0A E5 CD 2A : 72
4510 45 C1 CD 33 47 18 EF FE : 52
4518 2D 20 0C E5 CD 2A 45 4D : C7
4520 44 E1 CD 3A 47 18 DF 4D : B7
4528 44 C9 13 CD 84 45 CD 53 : D6
4530 43 FE 2A 20 0A E5 CD 83 : CA
4538 45 C1 CD 43 47 18 EF FE : 62
4540 2F 20 0C E5 CD 83 45 4D : 22
4548 44 E1 CD 51 47 18 DF FE : 7F
4550 25 C0 E5 CD 83 45 4D 44 : F0
4558 E1 D5 CD 5F 47 D1 18 CE : E0
4560 CD 83 45 7C 2F 67 7D 2F : 53
4568 6F 23 C9 4F 13 1A FE 0D : E2
4570 CA AD 48 FE 5E CC 1A 43 : 44
4578 26 00 6F 13 1A 13 B9 C2 : 50
SUM: 9C D6 8D AD FD D7 0D 87 D992

4588 AD 48 C9 13 CD 53 43 FE : 32
4598 28 CA 47 46 FE 2D CA 6D : 1D
4590 47 FE 24 CA 80 47 FE 25 : 1D
4598 CA 7B 47 CD 39 3A D2 86 : 24
45A0 47 CD 82 43 CA 8B 47 21 : 96
45A8 CB 4C 01 0C 00 ED B1 CA : 8C
45B0 AA 48 C5 3A 2E 62 A7 20 : 42
45B8 29 3A A5 62 FE 02 32 A5 : 39
45C0 62 CD 59 46 28 10 CD B8 : 8B
45C8 39 38 04 FE 01 28 39 3C : 13
45D0 03 32 A5 62 18 32 3E 03 : C7
45D8 32 A5 62 CD 2E 3A 13 13 : 94
45E0 18 26 3A A5 62 F6 02 32 : A9
45E8 A5 62 CD 59 46 28 26 CD : 8E
45F0 B8 39 DA 8C 48 FE 01 28 : C6
45F8 13 FE 05 CA 38 46 3E 03 : 9F
SUM: 1D C1 B2 A2 09 E3 6C FC 8C15

4600 32 A5 62 3E 80 CD BA 35 : B3
4608 60 69 C1 C9 60 69 3E E7 : 41
4610 CD BA 35 C1 C9 CD 12 3A : 5F
4618 78 FE 20 D2 92 48 F6 20 : 58
4620 CD 9C 35 1A CD 36 3A 38 : 2D
4628 06 13 CD 9C 35 18 F4 3E : 01
4630 03 32 A5 62 13 13 C1 C9 : EC
4638 3E 03 32 A5 62 3E 80 CD : 05
4640 BA 35 CD 75 41 C1 C9 C5 : C1
4648 13 CD 03 45 60 69 C1 CD : 7F
4650 53 43 13 FE 29 C2 A7 48 : 81
4658 C9 D5 CD 2E 3A 1A 13 FE : FE
4660 23 20 04 1A 13 FE 23 D1 : 6E
4668 C9 13 1A CD B8 1F D2 C1 : 2D
4670 46 2A 39 62 3A A5 62 F6 : 42
4678 02 32 A5 62 3A 45 62 47 : 63
SUM: 08 53 FD E8 F5 F7 6C 29 5152

4680 3A 4E 62 A7 28 04 B8 C2 : 37
4688 AA 48 78 32 4E 62 C9 13 : 22
4690 1A CD B8 1F 30 2B 2A 39 : 7C
4698 62 3A 45 62 FE 01 C8 D5 : DF
46A0 3E 87 CD 9C 35 CD 5C 3A : C6
46A8 ED 5B 39 62 B7 ED 52 3E : 17
46B0 E7 CD BA 35 3E C1 CD 9C : 0B
46B8 35 D1 3E 03 32 A5 62 B7 : 37
46C0 C9 6F 26 00 13 1A CD B8 : 10
46C8 1F D8 29 29 29 B5 6F : BF
46D0 13 18 F2 21 00 00 13 1A : 6B
46D8 FE 5F 28 FA D6 30 D8 FE : 5B
46E0 02 D0 0F ED 6A 18 EF D5 : 14
46E8 D6 30 4F 13 CD 69 43 20 : 01
46F0 FA 1B 1A D1 FE 48 20 0C : 72
46F8 79 CD C1 46 1A FE 48 C2 : 6F
SUM: E5 C3 77 EB 61 EC 57 B0 FB1D

4700 AA 48 13 C9 FE 42 20 14 : CC
4708 79 FE 02 D2 AA 48 6F 26 : C0
4710 00 CD D6 46 1A FE 42 C2 : 05
4718 AA 48 13 C9 69 26 00 13 : 6A
4720 1A CD 39 3A D8 D6 30 4D : 85
4728 44 29 29 09 29 4F 06 00 : 1D
4730 09 18 EC 09 3E C0 CD 9C : 7D
4738 35 C9 B7 ED 42 3E C1 CD : B0
4740 9C 35 C9 3A A5 62 B6 02 : C3
4748 CC 46 48 3E C2 CD 9C 35 : F8
4750 C9 3A A5 62 E6 02 CC 59 : 17

4758 48 3E C3 CD 9C 35 C9 3A : EA
4760 A5 62 E6 02 CC 70 48 3E : B1
4768 C7 CD 9C 35 C9 3A A5 62 : 6F
4770 E6 02 CC 60 45 3E C6 CD : 2A
4778 9C 35 C9 CD D3 46 18 0E : A6
SUM: C4 8B 93 EE 3C 65 77 0A 8FA5

4780 CD 8F 46 38 09 C9 CD E7 : 60
4788 46 18 03 CD D6 45 3E E7 : 03
4790 CD BA 35 C9 AF 32 A5 62 : 6D
4798 32 4E 62 32 51 62 3A 2E : 2F
47A0 62 A7 CA FC 44 CD C7 47 : EE
47A8 CD 53 43 FE 2B 20 08 E5 : 99
47B0 CD C6 47 C1 09 18 F1 FE : AB
47B8 2D C0 E5 CD C6 47 44 4D : 3D
47C0 E1 B7 ED 42 18 E2 13 CD : A1
47C8 53 43 FE 2D CA 60 45 FE : 2E
47D0 24 CA 69 46 FE 25 CA D3 : 5D
47D8 46 CD 39 3A D2 E7 46 CD : 52
47E0 B2 43 CA 6B 45 21 CB 4C : 77
47E8 01 00 ED B1 CA 4A 48 : 61
47F0 3A A5 62 F6 02 32 A5 62 : 72
47F8 CD B8 39 DA 8C 48 A7 CA : DD
SUM: 63 6C 0B 9F E8 A1 11 00 03E7

4800 92 48 FE 05 CA 92 48 60 : E1
4808 69 47 3A 4E 62 A7 28 04 : 6D
4810 B8 C2 A4 48 78 32 4E 62 : C0
4818 C9 D5 F5 AF 32 05 63 01 : DD
4820 0A 05 11 05 63 CD 60 48 : FD
4828 C6 30 1B 12 10 F7 6B 62 : F7
4830 01 20 04 3E 30 BE 20 04 : 75
4838 71 23 10 F9 F1 B7 28 01 : 6E
4840 EB CD E5 1F D1 C9 05 EB : 16
4848 21 00 00 3E 10 29 EB 29 : AC
4850 EB 30 01 09 3D 20 F6 D1 : 49
4858 C9 D5 CD 70 48 EB D1 C9 : A8
4860 C5 AF 06 10 29 17 2C 91 : 87
4868 30 02 2D 81 10 F6 C1 C9 : 70
4870 78 B1 28 30 EB 21 00 00 : 8D
4878 3E 10 EB 29 EB ED 6A 1C : C0
SUM: 29 E2 0A 58 DF C1 12 9A A358

4880 ED 42 30 02 09 1D 3D 20 : E4
4888 F1 C9 AF 01 3E 01 01 3E : E8
4890 02 01 3E 03 01 3E 04 01 : 88
4898 3E 05 01 3E 06 01 3E 07 : CE
48A0 01 3E 08 01 3E 09 01 3E : CE
48A8 0A 01 3E 0B 01 3E 0C 01 : A0
48B0 3E 0D 01 3E 0E 01 3E 0F : E6
48B8 01 3E 10 32 F6 B2 ED 53 : 1E
48C0 FE 62 87 21 65 49 4F 06 : 0B
48C8 00 09 5E 23 56 CD E5 1F : B1
48D0 11 55 4A CD E5 1F CD FC : 4A
48D8 48 3A FD 62 47 CD DF 1F : F3
48E0 3E 5E CD 64 1F CD EE 1F : 56
48E8 2A 3D 62 23 22 3D 62 ED : 9A
48F0 5B 31 62 CD 49 38 ED 7B : A4
48F8 2F 62 B7 C9 D5 CD F1 1F : C3
SUM: B1 C3 E9 E0 DC 18 C6 ED CB67

4900 2A 33 62 AF CD 19 48 CD : 69
4908 F1 1F CD F1 1F ED 5B 31 : 66
4910 62 2A FE 62 B7 ED 52 20 : 02
4918 07 2A 7A 1F 7E 32 FD 62 : D9
4920 1A 13 FE 0D 28 0A FE 09 : 71
4928 CA F4 1F CC 35 49 18 E1 : 1A
4930 D1 CD EE 1F C9 CD F1 1F : 51
4938 2A 7A 1F 7E E6 07 20 F5 : 43
4940 C9 CD 9A 1F 23 C9 F5 79 : A9
4948 CD 9A 1F 23 78 CD 9A 1F : A7
4950 23 F1 C9 CD 94 1F 23 C9 : 49
4958 F5 CD 94 1F 47 23 CD 94 : 48
4960 1F 47 23 F1 C9 87 49 8E : A1
4968 49 9E 49 AB 49 B9 49 C8 : EE
4970 49 D8 49 E8 49 F6 49 FE : DE
4978 49 11 4A 1D 4A 29 4A 37 : B5
SUM: 05 E7 E6 66 50 83 BD FE 2E3F

4980 4A 38 4A 44 4A 4A 4A 53 : 41
4988 79 6E 74 61 78 00 55 6E : F7
4990 64 65 66 69 6E 65 64 20 : EF
4998 4C 61 62 65 6C 00 52 65 : 97
49A0 64 65 66 69 6E 69 74 69 : 4C
49A8 6F 6E 00 49 6C 6C 65 67 : CA
49B0 61 6C 20 4C 61 62 65 6C : CD
49B8 00 49 6C 6C 65 67 61 6C : BA
49C0 20 4F 70 63 6F 64 65 00 : 7A
49C8 49 6C 6C 65 67 61 6C 20 : DA
49D0 4F 70 65 72 61 6E 64 00 : C9
49D8 54 6F 6F 20 4D 61 6E 79 : E7
49E0 20 4C 61 62 65 6C 73 00 : 73
49E8 4D 69 73 73 69 6E 67 20 : FA
49F0 4C 61 62 65 6C 00 54 6F : A3
49F8 6F 20 46 61 72 00 49 6C : 5D
SUM: DB C4 A4 D2 6C BB 0E 82 813E

4A00 6C 65 67 61 6C 20 45 78 : E2
4A08 70 72 65 73 73 69 6F 6E : 73
4A10 00 4D 69 73 73 69 6E 67 : DA
4A18 20 5B 29 5D 00 4D 69 73 : 2A
4A20 73 69 6E 67 20 5B 2E 5D : B7
4A28 00 4D 69 73 73 69 6E 67 : DA
4A30 20 51 75 6F 74 65 00 00 : 2E
4A38 49 6C 65 67 61 6C 20 : DA
4A40 4F 52 47 00 56 61 6C 75 : 80

▶うちのパソコンはコタツの上に置いてあるけど、読者の皆さんはどこに置いてあるのてしょうか。ぜひ調べて発表してください。コタツでパソコンは腰にきます。

奥江 聡 (20) 神奈川県

4C00	44	ED	A3	ED	B3	ED	AB	ED	:	F5
4C08	BB	ED	6F	ED	67	ED	4D	ED	:	92
4C10	45	42	43	00	44	45	00	48	:	9E
4C18	4C	00	53	50	00	41	46	00	:	76
4C20	42	00	43	00	44	00	45	00	:	0E
4C28	48	00	4C	00	28	48	4C	29	:	79
4C30	00	41	00	49	58	00	49	49	:	84
4C38	00	28	49	58	00	28	49	59	:	93
4C40	00	00	4E	5A	00	5A	00	4E	:	50
4C48	43	00	43	00	50	4F	00	50	:	75
4C50	45	00	50	00	4D	00	00	28	:	0A
4C58	44	45	29	2C	41	00	12	00	:	31
4C60	28	42	43	29	2C	41	00	02	:	45
4C68	00	52	2C	00	ED	00	4F	49	:	24
4C70	2C	41	00	ED	47	00	44	45	:	2A

```
4E80 39 63 2A 64 1F CD 03 20 : 3
4E88 D8 CD 43 50 D8 06 10 21 : 4
```

SUM: 48 A9 99 40 49 51 48 18 693

THE SENTINEL


```

0104' C9                                RET
0105'
0105'
0105'
0105' F5                               pPRINT: PUSH AF
0106' C5                                 PUSH BC
0107' E5                                 PUSH DE
0108' E3                                 PUSH HL
0109'
0109' L1 10 **                           LD    DE,PENBUF
010C' 2A **                             LD    HL,(CURK)
010F' 26 00                            LD    L,H,*
0111' 19                                ADD    HL,DE
0112' 77                                LD    (HL),A
0113' 23                                INC    HL
0114' 36 00                            LD    (HL),A
0115'
0115' L0                                LD    A,(CURK)
0119' 3C                                INC    A
011A' 32 **                             LD    (CURK),A
011E' FE 16                             CP     Z
011F' D4 **                             CALL   NC,pPRINT1
0120'
0120' E1                                POP    HL
0121' D1                                POP    DE
0124' C1                                POP    BC
0125' F1                                POP    AF
0126' C9                                RET
0127'
0127' CD ** **                         pPRINT1:CALL PUTPT
01FA' C0 ** **                         pPRINT2:CALL pPENTS
01FD' C0 ** **                         CALL   GETADS
0200' CD ** **                         CALL   pPTH1
0203' CD ** **                         CALL   pPTS0
0204' CD ** **                         CALL   pPENT5
0205' C9                                RET
0206'
0206'                                     ; flush PEN-buffer
0206' 3A ** **                         PUTTY: LD  A,(pass)
020B' A7                                AND    R
020E' C6                                RET    Z
020F'
020F' L0 3A ** **                     LD      A,(prname)
0212' A7                                AND    Z
0213' C6                                RET    A
0214'
0214' 2A ** **                       LD      HL,(CNTLINE)
0217' 7C                                LD     LP,A,R
0218' B5                                OR     L
0219' CC ** **                       CALL    Z,NXTPG
021C'
021C' L0 2A ** **                     LD      HL,(CNTLINE)
021F' 23                                LD     HL,CNTLINE,HL
0220' 26 ** **                       LD     DEC HL
0223' 28
0224' 05                                PUSH   DE
0225' D0 58 ** **                     LD     HL,(PAGE)
0229' 87                                OR     R
022A' FD 52                          SRC    HL,DE
022C' 01                                POP    DE
022D' 38 05                          JR     C,pPT2
022F'
022F' L1 00 00                        LD      HL,*
0232' 23 ** **                       LD     (CNTLINE),HL;The last line
0235'
0235' CD ** **                       pPT2: CALL PUTEN
0236'                                     ;PUT Error mark (if exists)
0236'
0236' L0 21 ** **                     LD      HL,PENBUF
0239' 7E                                pPT2: LD  A,(HL)
023C' A7                                AND    A
023D' 26 00                            LD     Z,pPT4
023F' 23                                INC    HL
0240' CD ** **                       CALL    pTPN
0243' 16 78                            JR     pPT3
0245'
0245' 3A ** **                       pPT4: LD  A,(TXTFLAG)
0248' A7                                AND    A
0249' 38 05                            JR     Z,pPT5
024B' C0 ** **                       CALL    puExit
024E' 16 05                            JR     pPT6
0250'
0250' pPT5: LD  A,CR
0252' CD ** **                       CALL    putprn
0255' A7                                pPT6: XOR  A
0256' 32 ** **                       LD     (CURK),A
0259' 32 ** **                       LD     (TXTFLAG),A

```

```

0255C  C0                                RET
0255D  C0                                LD  Open Next Page
0256D  C1 00 00 NKTPO:                  LD  L,(CNTLINE),HL
0257D  C2 00 00                                LD  A,(TITLEBF)
0258D  C3 3A 00 00                                LD  AND  A
0259D  C4 00 00                                LD  RET  Z
0260D  C5                                LD  HL,TITLEF
0261D  C6 21 00 00                                LD  NKT1:  LD  A,(HL)
0262D  C7 20 00                                LD  INC  HL
0263D  C8 27 00 00                                LD  AND  A
0264D  C9 28 00                                LD  JR  NKT2
0265D  CA 00 00                                CALL  putprn
0266D  CB 18 00                                LD  A,(HL)
0267D  CC 3E 00                                LD  A,CR
0268D  CD 00 00                                LD  putprn
0269D  CE 27 00 00                                LD  HL,1
0270D  CF 22 00 00                                LD  (CNTLINE),HL
0271D  D0                                LD  RET1
0272D  D1                                LD  RET1
0273D  D2 3A 00 00 PUTEK:                LD  A,(ERRNO)
0274D  D3 00 00                                LD  AND  A
0275D  D4 C8 00                                LD  RET  Z
0276D  D5 21 00 00                                LD  HL,ERRTAB
0277D  D6 00 00                                LD  PUSH  DE
0278D  D7 18 00                                LD  D,0
0279D  D8 5F 00                                LD  L,A
0280D  D9 1D 00                                LD  ADD  HL,D
0281D  DA 00 00                                LD  LD  A,(HL)
0282D  DB 01 00                                LD  POP  DE
0283D  DC 33 00 00                                LD  (PRTNBUF),A :Put Err msg
0284D  DD AF 00                                LD  XOR  A
0285D  DE 00 00                                LD  (ERRNO),A :Clear (ERRNO)
0286D  DF                                LD  RET
0287D  E0                                LD  RET
0288D  E1                                LD  RET
0289D  E2 55 00 40 53 53                    ERRTBL:  DB  'SUNSSASEA'
0290D  E3 51 55 45 41                    DB  'SUNSSASEA'
0291D  E4 53 53 49 41                    DB  'SSSIAYRI'
0292D  E5 56 52 49                    DB  'SSSIAYRI'
0293D  E6                                LD  RET
0294D  E7                                LD  RET
0295D  E8 0F                                LD  RET
0296D  E9 0F                                LD  RET
0297D  EA 0F                                LD  RET
0298D  EB 0F                                LD  RET
0299D  EC 33 00 00                                LD  CALL  pprthx1
0300D  ED 01 00                                LD  POP  AF
0301D  EE 18 0F                                LD  pprthx1:  AND  0FH
0302D  EF 18 0F                                LD  AND  A,'8'
0303D  F0 3A 00                                LD  CP  '9'
0304D  F1 3A 00                                LD  CP  '9'
0305D  F2 3A 00                                LD  AND  A,'8'
0306D  F3 3A 00                                LD  CP  '9'
0307D  F4 3A 00                                LD  AND  A,'8'
0308D  F5 3A 00                                LD  CP  '9'
0309D  F6 3A 00                                LD  AND  A,'8'
0310D  F7 3A 00                                LD  CP  '9'
0311D  F8 3A 00                                LD  AND  A,'8'
0312D  F9 3A 00                                LD  CP  '9'
0313D  FA 3A 00                                LD  AND  A,'8'
0314D  FB 3A 00                                LD  CP  '9'
0315D  FC 3A 00                                LD  AND  A,'8'
0316D  FD 3A 00                                LD  CP  '9'
0317D  FE 3A 00                                LD  AND  A,'8'
0318D  FF 3A 00                                LD  CP  '9'
0319D  00 3A 00                                LD  AND  A,'8'
0320D  01 3A 00                                LD  CP  '9'
0321D  02 3A 00                                LD  AND  A,'8'
0322D  03 3A 00                                LD  CP  '9'
0323D  04 3A 00                                LD  AND  A,'8'
0324D  05 3A 00                                LD  CP  '9'
0325D  06 3A 00                                LD  AND  A,'8'
0326D  07 3A 00                                LD  CP  '9'
0327D  08 3A 00                                LD  AND  A,'8'
0328D  09 3A 00                                LD  CP  '9'
0329D  0A 3A 00                                LD  AND  A,'8'
0330D  0B 3A 00                                LD  CP  '9'
0331D  0C 3A 00                                LD  AND  A,'8'
0332D  0D 3A 00                                LD  CP  '9'
0333D  0E 3A 00                                LD  AND  A,'8'
0334D  0F 3A 00                                LD  CP  '9'
0335D  10 3A 00                                LD  AND  A,'8'
0336D  11 3A 00                                LD  CP  '9'
0337D  12 3A 00                                LD  AND  A,'8'
0338D  13 3A 00                                LD  CP  '9'
0339D  14 3A 00                                LD  AND  A,'8'
0340D  15 3A 00                                LD  CP  '9'
0341D  16 3A 00                                LD  AND  A,'8'
0342D  17 3A 00                                LD  CP  '9'
0343D  18 3A 00                                LD  AND  A,'8'
0344D  19 3A 00                                LD  CP  '9'
0345D  1A 3A 00                                LD  AND  A,'8'
0346D  1B 3A 00                                LD  CP  '9'
0347D  1C 3A 00                                LD  AND  A,'8'
0348D  1D 3A 00                                LD  CP  '9'
0349D  1E 3A 00                                LD  AND  A,'8'
0350D  1F 3A 00                                LD  CP  '9'
0351D  20 3A 00                                LD  AND  A,'8'
0352D  21 3A 00                                LD  CP  '9'
0353D  22 3A 00                                LD  AND  A,'8'
0354D  23 3A 00                                LD  CP  '9'
0355D  24 3A 00                                LD  AND  A,'8'
0356D  25 3A 00                                LD  CP  '9'
0357D  26 3A 00                                LD  AND  A,'8'
0358D  27 3A 00                                LD  CP  '9'
0359D  28 3A 00                                LD  AND  A,'8'
0360D  29 3A 00                                LD  CP  '9'
0361D  2A 3A 00                                LD  AND  A,'8'
0362D  2B 3A 00                                LD  CP  '9'
0363D  2C 3A 00                                LD  AND  A,'8'
0364D  2D 3A 00                                LD  CP  '9'
0365D  2E 3A 00                                LD  AND  A,'8'
0366D  2F 3A 00                                LD  CP  '9'
0367D  30 3A 00                                LD  AND  A,'8'
0368D  31 3A 00                                LD  CP  '9'
0369D  32 3A 00                                LD  AND  A,'8'
0370D  33 3A 00                                LD  CP  '9'
0371D  34 3A 00                                LD  AND  A,'8'
0372D  35 3A 00                                LD  CP  '9'
0373D  36 3A 00                                LD  AND  A,'8'
0374D  37 3A 00                                LD  CP  '9'
0375D  38 3A 00                                LD  AND  A,'8'
0376D  39 3A 00                                LD  CP  '9'
0377D  3A 3A 00                                LD  AND  A,'8'
0378D  3B 3A 00                                LD  CP  '9'
0379D  3C 3A 00                                LD  AND  A,'8'
0380D  3D 3A 00                                LD  CP  '9'
0381D  3E 3A 00                                LD  AND  A,'8'
0382D  3F 3A 00                                LD  CP  '9'
0383D  40 3A 00                                LD  AND  A,'8'
0384D  41 3A 00                                LD  CP  '9'
0385D  42 3A 00                                LD  AND  A,'8'
0386D  43 3A 00                                LD  CP  '9'
0387D  44 3A 00                                LD  AND  A,'8'
0388D  45 3A 00                                LD  CP  '9'
0389D  46 3A 00                                LD  AND  A,'8'
0390D  47 3A 00                                LD  CP  '9'
0391D  48 3A 00                                LD  AND  A,'8'
0392D  49 3A 00                                LD  CP  '9'
0393D  4A 3A 00                                LD  AND  A,'8'
0394D  4B 3A 00                                LD  CP  '9'
0395D  4C 3A 00                                LD  AND  A,'8'
0396D  4D 3A 00                                LD  CP  '9'
0397D  4E 3A 00                                LD  AND  A,'8'
0398D  4F 3A 00                                LD  CP  '9'
0399D  50 3A 00                                LD  AND  A,'8'
0400D  51 3A 00                                LD  CP  '9'
0401D  52 3A 00                                LD  AND  A,'8'
0402D  53 3A 00                                LD  CP  '9'
0403D  54 3A 00                                LD  AND  A,'8'
0404D  55 3A 00                                LD  CP  '9'
0405D  56 3A 00                                LD  AND  A,'8'
0406D  57 3A 00                                LD  CP  '9'
0407D  58 3A 00                                LD  AND  A,'8'
0408D  59 3A 00                                LD  CP  '9'
0409D  5A 3A 00                                LD  AND  A,'8'
0410D  5B 3A 00                                LD  CP  '9'
0411D  5C 3A 00                                LD  AND  A,'8'
0412D  5D 3A 00                                LD  CP  '9'
0413D  5E 3A 00                                LD  AND  A,'8'
0414D  5F 3A 00                                LD  CP  '9'
0415D  60 3A 00                                LD  AND  A,'8'
0416D  61 3A 00                                LD  CP  '9'
0417D  62 3A 00                                LD  AND  A,'8'
0418D  63 3A 00                                LD  CP  '9'
0419D  64 3A 00                                LD  AND  A,'8'
0420D  65 3A 00                                LD  CP  '9'
0421D  66 3A 00                                LD  AND  A,'8'
0422D  67 3A 00                                LD  CP  '9'
0423D  68 3A 00                                LD  AND  A,'8'
0424D  69 3A 00                                LD  CP  '9'
0425D  6A 3A 00                                LD  AND  A,'8'
0426D  6B 3A 00                                LD  CP  '9'
0427D  6C 3A 00                                LD  AND  A,'8'
0428D 
```

```

020A' C5
0220'
0220' E5
0222' 21 == **
0222' E2
0224' 20
0224' E1
0222' C8
0222' C8
0223'
0223' F5
0224' C5
0225' 2E 2A
0227' C0 == **
0227' 2E 2A
022C' C0 == **
022C' 18 F1
0284' C1
0285' F1
0286' C0
0287'
0287'
0287' : Print Text into PEN-file
0287'
0287' 2A == ** puttst: LD A,(CURX)
028A' FE 86 CP 8
028C' DC == ** C,putTAB
028C' FE 18 CP 18
0211' DC == ** CALL C,putTAB
0214'
0214' 3A == ** LD A,(psname)
0217' A7 AND A
0218' 28 23 JR Z,PTT2
021A' 3A == ** LD A,(fivl)
021B' A7 AND A
021C' 28 10 JR Z,PTT2
021E'
0220' 3A == ** LD A,(CURX)
0223' 47 LD B,A
0224' 3E 16 LD A,Z
0227' 68 SUB B
0227' 47 LD B,A
0228' C5 PTT1: PUSH BC
0228' 2E 28 LD A,
0228' C0 == ** CALL putTAB
022C' C1 POP BC
022C' 18 F7 PUSH PTT1
0231' 2E 43 LD A,C
0232' 3A == ** CALL putTAB
0232' 3E 28 LD A,
0236' C0 == ** CALL putTAB
0236' 18 86 JR PTT3
0237'
0237' 3A == ** PTT2: LD A,(CURX)
0240' FE 15 CP 24
0242' C0 == ** CALL C,putTAB
0245'
0245' 11 == ** PTT3: LD DE,shuf
0246' 1A PTT4: LD A,(DE)
0246' 13 INC DE
0246' A7 AND A
0246' C8 RET Z
024C' C0 == ** CALL putTAB
024F' 18 F7 JR PTT4
0251'
0251' F5 puttAB: PUSH AF
0252' 3E 80 LD A,TAB
0254' C0 == ** CALL putTAB
0254' 3A == ** LD A,(CURX)
025A' C8 86 ADD A,B
025C' 3E F5 AND AFH
025E' 32 == ** LD LD (CURX),A
0261' F1 POP AF
0262' C0 RET
0263'
0263' :
0263' : Works
0263'
0263' DSEG
0264'
0000' 00 CURX: DB 0
0001' 00
0002' 00
0003' 00
0004' 00
0005' 00
0006' 00
0007' 00
0008' 00
0009' 00
0010' 00
0011' 00
0012' 00
0013' 00
0014' 00
0015' 00
0016' 00
0017' 00
0018' 00
0019' 00
0020' 00
0021' 00
0022' 00
0023' 00
0024' 00
0025' 00
0026' 00
0027' 00
0028' 00
0029' 00
0030' 00
0031' 00
0032' 00
0033' 00
0034' 00
0035' 00
0036' 00
0037' 00
0038' 00
0039' 00
0040' 00
0041' 00
0042' 00
0043' 00
0044' 00
0045' 00
0046' 00
0047' 00
0048' 00
0049' 00
0050' 00
0051' 00
0052' 00
0053' 00
0054' 00
0055' 00
0056' 00
0057' 00
0058' 00
0059' 00
0060' 00
0061' 00
0062' 00
0063' 00
0064' 00
0065' 00
0066' 00
0067' 00
0068' 00
0069' 00
0070' 00
0071' 00
0072' 00
0073' 00
0074' 00
0075' 00
0076' 00
0077' 00
0078' 00
0079' 00
0080' 00
0081' 00
0082' 00
0083' 00
0084' 00
0085' 00
0086' 00
0087' 00
0088' 00
0089' 00
0090' 00
0091' 00
0092' 00
0093' 00
0094' 00
0095' 00
0096' 00
0097' 00
0098' 00
0099' 00
0100' 00
0101' 00
0102' 00
0103' 00
0104' 00
0105' 00
0106' 00
0107' 00
0108' 00
0109' 00
0110' 00
0111' 00
0112' 00
0113' 00
0114' 00
0115' 00
0116' 00
0117' 00
0118' 00
0119' 00
0120' 00
0121' 00
0122' 00
0123' 00
0124' 00
0125' 00
0126' 00
0127' 00
0128' 00
0129' 00
0130' 00
0131' 00
0132' 00
0133' 00
0134' 00
0135' 00
0136' 00
0137' 00
0138' 00
0139' 00
0140' 00
0141' 00
0142' 00
0143' 00
0144' 00
0145' 00
0146' 00
0147' 00
0148' 00
0149' 00
0150' 00
0151' 00
0152' 00
0153' 00
0154' 00
0155' 00
0156' 00
0157' 00
0158' 00
0159' 00
0160' 00
0161' 00
0162' 00
0163' 00
0164' 00
0165' 00
0166' 00
0167' 00
0168' 00
0169' 00
0170' 00
0171' 00
0172' 00
0173' 00
0174' 00
0175' 00
0176' 00
0177' 00
0178' 00
0179' 00
0180' 00
0181' 00
0182' 00
0183' 00
0184' 00
0185' 00
0186' 00
0187' 00
0188' 00
0189' 00
0190' 00
0191' 00
0192' 00
0193' 00
0194' 00
0195' 00
0196' 00
0197' 00
0198' 00
0199' 00
0200' 00
0201' 00
0202' 00
0203' 00
0204' 00
0205' 00
0206' 00
0207' 00
0208' 00
0209' 00
0210' 00
0211' 00
0212' 00
0213' 00
0214' 00
0215' 00
0216' 00
0217' 00
0218' 00
0219' 00
0220' 00
0221' 00
0222' 00
0223' 00
0224' 00
0225' 00
0226' 00
0227' 00
0228' 00
0229' 00
0230' 00
0231' 00
0232' 00
0233' 00
0234' 00
0235' 00
0236' 00
0237' 00
0238' 00
0239' 00
0240' 00
0241' 00
0242' 00
0243' 00
0244' 00
0245' 00
0246' 00
0247' 00
0248' 00
0249' 00
0250' 00
0251' 00
0252' 00
0253' 00
0254' 00
0255' 00
0256' 00
0257' 00
0258' 00
0259' 00
0260' 00
0261' 00
0262' 00
0263' 00
0264' 00
0265' 00
0266' 00
0267' 00
0268' 00
0269' 00
0270' 00
0271' 00
0272' 00
0273' 00
0274' 00
0275' 00
0276' 00
0277' 00
0278' 00
0279' 00
0280' 00
0281' 00
0282' 00
0283' 00
0284' 00
0285' 00
0286' 00
0287' 00
0288' 00
0289' 00
0290' 00
0291' 00
0292' 00
0293' 00
0294' 00
0295' 00
0296' 00
0297' 00
0298' 00
0299' 00
0300' 00
0301' 00
0302' 00
0303' 00
0304' 00
0305' 00
0306' 00
0307' 00
0308' 00
0309' 00
0310' 00
0311' 00
0312' 00
0313' 00
0314' 00
0315' 00
0316' 00
0317' 00
0318' 00
0319' 00
0320' 00
0321' 00
0322' 00
0323' 00
0324' 00
0325' 00
0326' 00
0327' 00
0328' 00
0329' 00
0330' 00
0331' 00
0332' 00
0333' 00
0334' 00
0335' 00
0336' 00
0337' 00
0338' 00
0339' 00
0340' 00
0341' 00
0342' 00
0343' 00
0344' 00
0345' 00
0346' 00
0347' 00
0348' 00
0349' 00
0350' 00
0351' 00

```

THE SENTINEL

リスト4 WZDソースリスト3

[illegible]

```

0009      TAB      EQU    00H
0000      CR      EQU    00H
0020      SPC      EQU    ' '
0000      :
0000      :          START
0000      :
0000      :          .LIST
0000      :
0000      :          assemble a line
0000      :
0000      :
0000  00 73  **      linasm:LD      (SPBUF),SP
0000      :
0000      :
0000  0007  1A  **      LD      A,(prname)
0000  0007  37  **      AND    A
0000  0007  44  **      CALL   NZ,PUTH0
0000      :
0000  0000  CD  **      CALL   GETADRS
0000  0000  22  **      LD      (PADDR),R
0000  0011      :
0000  0011  2A  **      LD      HL,(LINNO)
0000  0011  23  **      INC    HL
0000  0015  22  **      LD      (LINNO),HL
0000  0016      :
0000  0016  11  **      LD      DE,atbuff
0000  0016  53  **      LD      (LINPTR),DE
0000  0017      :
0000  0017  00  **      CALL   DLABEL ;define label (if exists)
0000  0022  00  **      CALL   SPCUT
0000  0022  1A  **      LD      A,(OE)
0000  0022  00  **      CR      ' '
0000  0026  28  30  JR      Z,NXLIN
0000  0026  FE  00  CR      CR
0000  0026  28  30  JR      Z,NXLIN
0000  0026  FE  00  CR      ' '
0000  0030  28  44  LD      Z,WCON
0000  0032  00  41  SUB    'A'
0000  0034  0A  **      JP      C,ERR4
0000  0037  FE  1A  LD      A,(Z-'A'+1)
0000  0037  00  **      JC      NC,ERR4
0000  003C  07  **      ADD    A,A
0000  003C  4E  **      LD      C,A
0000  003C  00  00  LD      B,A
0000  0040  21  **      LD      HL,CPTBL
0000  0040  00  **      ADD    HL,AC
0000  0044  7E  **      LD      A,(HL)
0000  0045  23  **      INC    HL
0000  0045  06  **      LD      H,(HL)
0000  0047  0F  **      LD      L,A
0000  0048  33  **      LNASM: EX    DE,HL
0000  0048  00  **      CMAP   CS,AP
0000  004C  33  **      EX    DE,HL
0000  0050  0A  **      JP      C,ERR4
0000  0054  CD  **      CALL   SPCUT
0000  0055  06  **      EX    AP,AP'
0000  0054  CD  **      CALL   CENDCG
0000  0057      :
0000  0057  CD  **      CALL   SPCUT
0000  005A  1A  **      LD      A,(OE)
0000  005A  37  **      AND    A
0000  005C  28  0F  JR      Z,ZLNASK2
0000  005C  FE  00  CR      CR
0000  0064  28  00  JR      Z,ZLNASK2
0000  0062  FE  30  CR      ' '
0000  0062  FE  30  CR      ' '
0000  0067      :
0000  0067      :          NZ,ERR4 ;SyntaxError
0000  0067      :
0000  0067  1A  **      XLIN:LD      A,(OE)
0000  0068  13  **      INC    DE
0000  0069  FE  00  CR      CR
0000  0069  20  FA  JR      NZ,XLIN
0000  0069      :
0000  0069  2A  **      LNASK2:LD      A,(prname)
0000  0074  47  **      AND    A
0000  0074  00  **      CR      ' '
0000  0074  00  **      CALL   SPCUT

```

```

0070: B7          ;
0070: C0          RET
0070:             ;
0070: 21 ** **   @CON: LD      HL,WTBL
0070: 10 C0       JR      LNASH1
0070:             ;
0070:             ; Define Labels (called from linasm)
0070: 1A          ;
0070: 1A          ;
0070: 1A          ;
0070: FE 40       ;
0070:             ;
0070: D8          ;
0070: CD ** **   CALL  SKIPLB ;Skip label strings
0070: CD ** **   CALL  SPICIT ;
0070: 21 ** **   LD      HL,SOBLX
0070: CD ** **   LD      @match
0070: 20 10       JR      Z,LAB1
0070: 21 ** **   LD      HL,SEQU
0070: CD ** **   LD      @match
0070: 20 40       JR      Z,LAB2
0070: 1A          LD      A,(DE)
0070: 13          INC     DE
0070: FE 3A       CP      '-'
0070: CA ** **   JP      Z,LAB3
0070: C0 ** **   JP      ERROR ;Syntax Error
0070:             ;
0070: 3A 3A 00 SDBL: DB      ":",'"',0
0070: 45 51 55 00: DB      DB,"'ENV",0
0070:             ;
0070:             ;
0070:             ;
0070: 3A ** **   LAB1: LD      A,(pass)
0070: A7          AND     A
0070: 50 33       JR      Z,LAB3
0070:             ;
0070: 3E 07       @AC: LD      A,07H ;For PASS-2
0070: CD ** **   CALL  PUTFA
0070: 50 50 ** ** LD      DE,(LINPR)
0070: CD ** **   CALL  STRLBI
0070: DA ** **   JP      C,ERR1 ;Undefined Label
0070: 3E 01       LD      A,01H
0070: 3E 01       CALL  PUTFAL
0070:             ;
0070: 50 50 ** ** LD      DE,(LINPR)
0070: CD ** **   CALL  LENDLB
0070: 70         LD      A,0
0070: FE 24       CP      31H
0070: CD ** **   JP      NC,ERR2 ;Too Long Label Name
0070:             ;
0070: CD ** **   CALL  PUTFAL
0070:             ;
0070: 1A          ;
0070: CD ** **   LAB2: LD      A,(DE)
0070: CD ** **   CALL  ISLENOW
0070: 30 00       JR      C,LAB2
0070: CD ** **   LD      INC DE
0070: CD ** **   CALL  PUTFA
0070: 10 F4       JR      LAB1
0070:             ;
0070: CD ** **   LAB2: INC     DE
0070: 13          RET
0070:             ;
0070: CD ** **   LAB3: CALL  GETADRS ;For PASS-1
0070: 44          LD      B,H
0070: 40          LD      C,L
0070: CD ** **   CALL  GETSEG
0070: 50 50 ** ** LD      DE,(LINPR)
0070: 21 ** **   LD      HL,(LENLOW)
0070: 23          INC     HL
0070: 23          LD      (LENLOW),HL
0070: 23          LD      (OFF),HL

```


[illegible]

0725:	10 05		JR	CALL4	
0727:	3E CD		CALL2:	JR	:CALL nn
0729:	CD ** *	CALL3:	CALL	PuToA	
0730:	CD ** *	CALL4:	CALL	EVAL	
0732:	CD		CALL	PuTnB2	
0733:				RET	
0733:	CD ** *	RET:	CALL	ISBLKX	
0736:	3E CD		LD	A,WCNH	:RET
0738:	CA ** *		JP	Z,PuToA	
0738:	CD ** *		CALL	GETCND	
073E:	DA ** *		JP	C,ERRS	
0741:	WE CD		LD	C,WCNH	:11 XXX 000
0741:	CD ** *		LD	PuSH2	
0746:					
0748:	CD ** *	IN:	CALL	GETRES	
0749:	DA ** *		C,ERRS		
074C:	CD ** *		CALL	CMACHK	
074F:	DE 05			SUB 5	
0751:	DA ** *		JP	C,ERRS	:FP
0752:	FE 06			CP 6	: (HL)?
0754:	CA ** *		JP	Z,ERRS	
0755:	FE 07			CP 7	:A?
0758:	2B 14			JR	Z,1NA
075D:	DE ** *		JP	NC,ERRS	
075E:	47		INC:	LD	S,A
0761:	CD ** *		CALL	PuTSD	
0764:	78			LD	A,6
0765:	05 48			LD	C,48H
0767:	CD ** *		CALL	PuSH2	:01 XXX 000
076A:	CD ** *		CALL	IS[C]	
076B:	CD			RET Z	
076E:	CD ** *		JP	ERRS	
0771:					
0771:	21 ** *	INA:	LD	HL[C]	
0774:	CD ** *		CALL	CMPGEN	
0777:	D8			RET NC	
0778:	3E CD			LD	A,WDH
077A:	CD ** *		CALL	PuToA	:IN A,(n)
077A:	CD ** *		CALL	EVAL	
0780:	CD ** *		CALL	PuTnB4	
0783:					
0783:	21 ** *	IS[C]:	LD	HL[C]	
0786:	08 03			LD	S,3
0789:	1A		IS[C]B:	LD	A,(DE)
078A:	CD			CP (HL)	
078A:	CD			RET NC	
078B:	13			INC DE	
078C:	23			INC HL	
078D:	19 F8			BLNZ	IS[C]B
078E:					
078F:					
0790:					
0790:	2B 43 20 00		[C]:	DEFS	'(C)',0
0794:	ED 76			IN	A,(C)
0795:	00			DEFS 0	
0797:					
0797:	D5		OUT:	PUSH DE	
0798:	CD ** *		CALL	IS[C]	
0798:	2B 1A			JR	Z,OUT(C)
0799:	CD			POP DE	
079E:	3E D3			LD	A,WDH
07A0:	CD ** *		CALL	PuToA	:OUT (n),A
07A3:	CD ** *		CALL	IS[C]A	
07A5:	CD ** *		CALL	CMACHK	
07A8:	CD ** *		CALL	GETRES	
07AC:	DA ** *		CALL	C,ERRS	
07AF:	FE 07			CP 7	:A?
07B1:	C2 ** *		JP	NZ,ERRS	
07B4:	CD ** *		JP	PuToA	
07B7:					
07B7:	C1		OUT[C]:	POP BC	
07B8:	CD ** *		CALL	PuTSD	
07B9:	CD ** *		CALL	CMACHK	
07B9:	CD ** *		CALL	GETRES	
07C1:	DA ** *		JP	C,ERRS	
07C4:	FE 06			CP 6	
07C8:	0E 41			LD	C,41H
07CB:	CD ** *		JP	NZ,PuSH2	:A
07CC:				ERRS	:XXX OUT (C),(HL)
07CE:					
07CE:	FE 06		RES:	LD	C,68H
07D0:	21			DEFS 21H	:10 XXX XXX
07D1:	FE CD			LD	C,WCNH
07D2:	21		SRT:	DEFS 21H	:11 XXX XXX
07D4:	FE 40			LD	C,40H
07D6:	1A			LD	A,(DE)
07D7:	CD ** *		CALL	ISDEC	:01 XXX XXX
07D8:	DA ** *		JP	C,ERRS	
07D9:	FE 38			CP "7"+1	:0-7?
07D9:	D2 ** *		JP	NC,ERRS	
07E2:	DE 30			SUB 0	
07E4:	07			ADD A,A	
07E5:	07			ADD A,A	
07E6:	07			ADD A,A	
07E7:	B1			OR C	
07E8:	4F			LD	C,A
07E9:	13			INC DE	
07EA:	CD ** *		CALL		


```

00000: D5 ** ** JP NC,ERR3 ;Too Long Label Err
00001: 0000: F8 28 OR Z,24H
00002: 0000: C0 ** ** CALL PUTFA
00003: 0000:
00004: 0000: 1A ** ** EXTJ: LD A,(DE)
00005: 0000: C0 ** ** CALL ISALNUM
00006: 0000: 38 06 JR C,EXT4
00007: 0000: 15 ** ** INC DE
00008: 0000: C0 ** ** CALL PUTFA
00009: 0000: 16 74 JR EXT3
00010: 0001:
00011: 0001: 31 61 EXTJ: LD A,B18H
00012: 0001: C0 ** ** CALL PUTFAHL
00013: 0001: 22 ** ** LD HL,(SV_STR)
00014: 0001: 23 ** ** INC HL
00015: 0001: 0000: AT ** ** INC HL
00016: 0001: C0 9A 1F ** ** XOR A
00017: 0001: 0000: 00 ** ** CALL _Poke
00018: 0001: 0000: 00 ** ** RET
00019: 0001:
00020: 0002: ** *_PUBLIC: LD A,(pass)
00021: 0002: 0000: 00 ** ** AND A
00022: 0002: C4 ** ** JP Z,NKLin
00023: 0002:
00024: 0002: 00 ** *_PUB1: CALL SPUT
00025: 0002: C4 05 ** ** PUSH DE ;For PASS-2
00026: 0002: C0 00 ** ** CALL SEALBI
00027: 0002: 0000: 00 ** ** JP C,ERR1 ;Undefined Label Flag
00028: 0002: 01 ** ** POP DE
00029: 0002:
00030: 0002: FE 01 CP J
00031: 0002: D4 13 ** ** JR NZ,PUB4
00032: 0002: D6 08 ** ** LD H,B
00033: 0002: D7 65 ** ** LD L,C
00034: 0002: 3E 87 ** ** LD A,WEH
00035: 0002: C0 ** ** CALL PUTFAHL
00036: 0002: 24 ** ** LD HL,(BLNUM)
00037: 0002: 23 ** ** INC HL
00038: 0002: 22 ** ** LD (BLNUM),HL
00039: 0002: 3E 01 ** ** LD A,B18H
00040: 0002: C0 ** ** CALL PUTFAHL
00041: 0002:
00042: 0002: 00 ** *_PUB4: CALL LEMBI
00043: 0002: 78 ** ** LD A,B
00044: 0002: FE 28 ** ** CP 31+1
00045: 0002: 0000: 00 ** ** JP NC,ERR3 ;Too Long Label Name
00046: 0002:
00047: 0002: C0 ** ** CALL PUTFAHL
00048: 0002: 1A ** ** LD A,(DE)
00049: 0002: C0 ** ** CALL ISALNUM
00050: 0002: 38 06 JR C,PUB3
00051: 0002: 15 ** ** INC DE
00052: 0002: C0 ** ** CALL PUTFA
00053: 0002: 16 74 JR PUB2
00054: 0002: 0000: 00 ** *_PUB3: CALL ISSPFE
00055: 0002: 0000: 00 ** ** LD A,PUB1
00056: 0002: 0000: 00 ** ** RET
00057: 0002:
00058: 0002: 3E 01 ;
00059: 0002: 32 ** ** LD A,1
00060: 0002: 0000: 00 ** ** (endf1g),A
00061: 0002:
00062: 0002: C0 ** ** CALL EOLCHK
00063: 0002: C0 ** ** RET Z
00064: 0002: 1A ** ** CALL EVAL
00065: 0002: 13 ** ** CALL PUTef
00066: 0002: 3E 19 ** ** LD A,WEH
00067: 0002: 0000: 00 ** ** CALL PUTFA
00068: 0002: 19 ** ** RET
00069: 0002:
00070: 0002:
00071: 0002: 3A ** ** *_CSRG: LD A,(LOCFLG)
00072: 0002: FE 02 ** ** AND 2
00073: 0002: C2 01 ** ** JP NZ,ERR16 ;Not Enable between PHASE
00074: 0002: 3E 02 ** ** LD A,2
00075: 0002: 3E 06 ** ** LD (CSRGD),A
00076: 0002: 3A ** ** LD A,(LOCOD)
00077: 0002: C2 32 ** ** LD A,(LOCFLG),A
00078: 0002: 2A ** ** LD HL,(PTRFC)
00079: 0002: 22 ** ** LD (PTRFC),HL
00080: 0002: 3E 01 ** ** LD A,WEH
00081: 0002: C0 ** ** CALL PUTFA
00082: 0002: 3A ** ** LD A,0
00083: 0002: C0 ** ** CALL PUTFA
00084: 0002: 39 ** ** RET
00085: 0002:
00086: 0002:
00087: 0002: 3A ** ** *_DSRG: LD A,(LOCFLG)
00088: 0002: FE 02 ** ** AND 2
00089: 0002: C2 01 ** ** JP NZ,ERR16 ;Not Enable between PHASE
00090: 0002: 3E 03 ** ** LD A,3
00091: 0002: 3A ** ** LD (CSRGD),A
00092: 0002: 3A ** ** LD A,(LOCOT)
00093: 0002: C2 32 ** ** LD A,(LOCFLG),A
00094: 0002: 2A ** ** LD HL,(PTRFC)
00095: 0002: 22 ** ** LD (PTRFC),HL
00096: 0002: 3E 01 ** ** LD A,WEH
00097: 0002: C0 ** ** CALL PUTFA
00098: 0002: 3A ** ** LD A,1
00099: 0002: C0 ** ** CALL PUTFA
00100: 0002: 39 ** ** RET
00101: 0002:
00102: 0002:
00103: 0002: 3A ** ** *_COMMON: LD A,(LOCFLG)
00104: 0002: FE 02 ** ** AND 2
00105: 0002: C2 01 ** ** JP NZ,ERR16 ;Not Enable between PHASE
00106: 0002: 3E 04 ** ** LD A,4
00107: 0002: 3A ** ** LD (CSRGD),A
00108: 0002: 3A ** ** LD A,(LOCOT)
00109: 0002: C2 32 ** ** LD A,(LOCFLG),A
00110: 0002: 2A ** ** LD HL,(PTRFC)
00111: 0002: 22 ** ** LD (PTRFC),HL
00112: 0002: 3E 01 ** ** LD A,WEH
00113: 0002: C0 ** ** CALL PUTFA
00114: 0002: 3A ** ** LD A,2
00115: 0002: C0 ** ** CALL PUTFA
00116: 0002: 39 ** ** RET
00117: 0002:
00118: 0002:
00119: 0002: C0 ** ** *_DEFB: CALL SPUT
00120: 0002: 1A ** ** LD A,(DE)
00121: 0002: C0 ** ** CALL ISQ
00122: 0002: 24 05 ** ** JR NZ,DEFB0
00123: 0002: 0000: 00 ** ** DMO
00124: 0002: 18 06 JR DEFB2
00125: 0002:
00126: 0002:
00127: 0002: C0 ** ** *_DEFB0: CALL EVAL
00128: 0002: C0 ** ** CALL PUTB04
00129: 0002: C0 ** ** *_DEFB2: CALL ISSPFE
00130: 0002: 2E 07 ** ** JR Z,_DEFB
00131: 0002: 0000: 00 ** ** RET
00132: 0002:
00133: 0002:
00134: 0002: C0 ** ** *_DEFW: CALL SPUT
00135: 0002: C0 ** ** CALL EVAL
00136: 0002: 3A ** ** LD A,(UNDEF)
00137: 0002: 28 01 ** ** AND 1
00138: 0002: 28 04 ** ** JR NZ,_DEFW1
00139: 0002: 0000: 00 ** **
00140: 0002:
00141: 0002: 78 ** ** LD A,C
00142: 0002: C0 ** ** CALL PUTFA
00143: 0002: 78 ** ** LD A,B
00144: 0002: C0 ** ** CALL PUTFA
00145: 0002: 18 00 JR _DEFW2
00146: 0002:
00147: 0002:
00148: 0002: C0 ** ** *_DEFW1: CALL PUTef
00149: 0002: C0 ** ** LD A,WEH
00150: 0002: C0 ** ** CALL PUTFA
00151: 0002: C0 ** ** *_DEFW2: CALL ISSPFE
00152: 0002: 28 0C ** ** JR Z,_DEFW
00153: 0002: C0 ** ** RET
00154: 0002:
00155: 0002:
00156: 0002: 7A ** ** *_DM: LD A,(DE)
00157: 0002: C0 ** ** CALL ISQ
00158: 0002: C0 ** ** *_DM2: LD A,ERR12
00159: 0002: C0 ** ** DMO: LD A,(KNJSJ)
00160: 0002: 40 17 ** ** AND A
00161: 0002: 28 20 ** ** JR NZ,DM00 ;No
00162: 0002: 0000: 00 ** ** put string if K-
00163: 0002:
00164: 0002: 1A ** ** DMO: INC DE
00165: 0002: 1A ** ** LD A,(DE)
00166: 0002: FE 80 ** ** CP C
00167: 0002: C0 ** ** JP Z,ERR12
00168: 0002: C0 ** ** LD A,ERR12
00169: 0002: 28 18 ** ** JR Z,DM04
00170: 0002: C0 ** ** CALL ISQ
00171: 0002: 28 02 ** ** JR NZ,DM3
00172: 0002: 40 13 ** ** INC DE
00173: 0002: 4F ** ** LD C,A
00174: 0002: 1A ** ** LD A,(DE)
00175: 0002: C0 ** ** JR Z,DM2
00176: 0002: 28 04 ** ** LD A,ERR12
00177: 0002: C0 ** ** CALL ISSPFE
00178: 0002: 0000: 00 ** ** RET Z
00179: 0002:

```

```

0AF2: 70                LD      A,C
0AF3: CD *** DMC3:    CALL  PUT0A
0AF5: 10 D0            JP      DM1
0AF7: CD *** DM4:    CALL  ESC
0AF8: 10 F0            JP      DM3
0AF9:                ; put string if #
0AFB:                ;
0AFD: 13            DMC#: INC  DE
0AFE: 1A            LD      A,(DE)
0AF9: FE 80          CP      CF
0A98: CA ***       JP      Z,ERR12
0A9A: FE 02          CP      '-'
0A95: 28 2C         JR      Z,DMK5
0A97: CD ***         CALL  ISLNJ
0A9A: 38 8F          JR      C,DMK2
0A98: CD ***         CALL  PUT0A
0A9E: 13            INC  DE
0A9F: 1A            LD      A,(DE)
0B01: FE 80          CP      CF
0B03: CA ***       JP      Z,ERR12
0B05: CD ***       CALL  PUT0A
0B07: 10 E1          JP      DMK0
0B08: CD *** DMK2:  CALL  ISQ
0B09: 28 FE          JR      NZ,DMK4
0B0A: 13            INC  DE
0B0B: 4F            LD      C,A
0B0C: 1A            LD      A,(DE)
0B0D: CD ***       CALL  ISQ
0B0E: 28 84          JR      Z,DMK3
0B0F: CD ***       CALL  ISLSPE
0B10: 18            RET  Z
0B11: CD          DMK3: DEC  DE
0B12: 19            LD      A,C
0B13: CD *** DMK4:  CALL  PUT0A
0B14: 18 8F          JR      DMK0
0B15: CD ***       CALL  ESC
0B16: 10 F6          JR      DMK4
0B17:                ;
0B18: 13            ESC#: INC  DE
0B19: 1A            LD      A,(DE)
0B1A: FE 80          CP      CF
0B1C: CA ***       JP      Z,ERR12
0B1D: D6 40          SUB  '#
0B1E: 3C 52          JR      C,ESC0
0B1F: FE 1C          CP      '['-'*'
0B20: D6          RET  C
0B21: CD 40          ESC0: ADD  A,'#
0B22: C9            RET
0B23:                ;
0B24: CD *** BEFS:   CALL  FVAL
0B25: CD ***       CALL  PUT0F
0B26: 3E D8          LD      A,BB#H
0B27: CD ***       CALL  PUTFA
0B28:                ;
0B29: 3A ***       LD      A,(UNDEF)
0B2A: FE 81          AND  1
0B2B: CD ***       JP      NZ,ERR15 ;Value Err
0B2C:                ;
0B2D: 60            LD      N,B
0B2E: 69            LD      L,C
0B2F: CD ***       CALL  ADDPC
0B30: C9            RET
0B31:                ;
0B32:                ; check whether kanji list char
0B33:                ;
0B34: FE 01          ISLN#: CP  01H
0B35: D6            RET  C
0B36: FE A8          CP      8AH#H
0B37: 3F            CCF
0B38: D6            RET  NC
0B39: FE E8          CP      8EH#H
0B3A: D6            RET  C
0B3B: FE F9          CP      0FH#H
0B3C: 3F            CCF
0B3D: C9            RET
0B3E:                ;
0B3F:                ; skip white space
0B40:                ;
0B41: 13            SPC0#: INC  DE
0B42: 1A            LD      A,(DE)
0B43: FE 00          SPC0F: CP  TAB
0B44: 26 FA          JR      Z,SPC0FH
0B45: 26 F8          CP      SPC
0B46: C9            JR      Z,SPC0FH
0B47: C9            RET
0B48:                ;
0B49:                ; check whether separator
0B50:                ;
0B51: CD *** ISSPRT: CALL  SPCUT
0B52: FE 2C          CP      ','
0B53: 26 83          JR      Z,ISSPRT1
0B54: FE 2A          CP      ' '
0B55: C9            RET  NZ
0B56:                ;
0B57: 13            ISSPRT1: INC DE
0B58: C9            RET
0B59:                ;
0B5A: 1A            ISSPRT2: LD  A,(DE)
0B5B: D6            ISLSPE: LD  A,(DE)
0B5C: 21 ***       ISLSPE: HL,ISPTBL
0B5D: 81 ***       LD  BC,SPEND1SPTBL
0B5E: D6 81        CP1R
0B5F: D9            X1X
0B60: C9            RET
0B61:                ;
0B62:                ; check whether blank
0B63:                ;
0B64: 1A            ISBLK#: LD  A,(DE)
0B65: FE 80          ISBLK: CP  TAB
0B66: C9            RET  Z
0B67: FE 80          CP      CR
0B68: C9            RET  Z
0B69: FE 3B          CP      '-'
0B6A: FE 20          RET  Z
0B6B: C9            CP      SPC
0B6C:                ;
0B6D:                ; check whether quotation mark
0B6E:                ;
0B6F: FE 27          ISQ#: CP  '-'
0B70: C9            RET  Z
0B71: FE 22          CP      '-'
0B72: C9            RET
0B73:                ;
0B74:                ; get condition code no
0B75:                ;
0B76: 21 ***       GETCD#: LD  HL,CNDTBL
0B77: 80 80          LD  B,8
0B78: 10 13          JR      GTREQ0
0B79:                ;
0B7A: CD *** REGCHK: CALL  GETREQ
0B7B: D6            RET  NC
0B7C: 31 ***       JP      ERR5
0B7D:                ;
0B7E:                ; get register no
0B7F:                ;
0B80: 21 ***       GETREG: LD  HL,REGTBL
0B81: 80 11          LD  B,17
0B82: 10 85          JR      GTREQ0
0B83:                ;
0B84:                ; get 8bit register no
0B85:                ;
0B86: 21 ***       GETREG: LD  HL,REGTBL
0B87: 80 80          LD  B,8
0B88: C9            GTREQ0: PUSH BC
0B89: FE 48          LD  C,B
0B8A: C5            GTREQ1: PUSH DE
0B8B: C9            CP      A,(OE)
0B8C: 24 DE          CP      (DE)
0B8D: 20 84          JR      NZ,GTREQ3
0B8E: 13            INC  DE
0B8F: C9            HL  HL
0B90: CD F6          JR      GTREQ2
0B91: C9            GTREQ3: (HL) (HL)
0B92: 35            DEC  (HL)
0B93: CD 8A          JR      NZ,GTREQ4
0B94: FE 80 ***     CALL  ISLSPE
0B95: 20 85          JR      NZ,GTREQ4
0B96: E1            POP  HL
0B97: 70            LD  A,C
0B98: 60            SUB  B
0B99: C9            POP  BC
0B9A: CD          RET
0B9B: CD *** GTREQ4: CALL  SRAZ
0B9C: D1            POP  DE
0B9D: 10 E3          DJNZ GTREQ1
0B9E: C9            POP  BC
0B9F: 37            SCF
0BA0: C9            RET
0BA1:                ;
0BA2: 3A            CMPCN: INC  (HL)
0BA3:                ;

```



```

OC24'
OC24' C0 *** PUT(R0):CALL PUT(X)
OC27' 1A GET(XD: LD A,(DE)
OC28' 13 INC DE
OC29' FE 29 CP
OC2B' 28 10 JR Z,EVAL1
OC2D' FE 20 CP
OC2E' 28 EC JZ 2,GET(XD
OC31' FE 29 CP
OC33' C2 *** JP NZ,ERR5
OC34' AF JDB A,
OC37' 32 *** LD (UNDEF),A
OC3A' 00 00 LD C,0
OC3C' C0 RET
OC3D'
OC3D' :
OC3D' : evaluate 'expression'
OC3D' :
OC3D' :
OC3D' C0 *** EVAL: CALL EVAL
OC40' :
OC40' : LD A,(DE)
OC41' FE 29 CP
OC43' C2 *** JP NZ,ERR10
OC45' 13 INC DE
OC47' C0 RET
OC4B' :
OC4B' : evaluate 'expression'
OC4B' :
OC4B' :
OC4B' C0 *** [EVAL: CALL ISI
OC4C' C2 *** JP NZ,ERR5
OC4E' 10 ED JR EVAL
OC50' :
OC50' : check ','
OC50' :
OC50' :
OC50' 47 CMACR: LD B,A
OC51' 1A LD A,(DE)
OC52' 12 CP
OC54' 78 LD A,B
OC55' C2 *** JP NZ,ERR11
OC56' 13 INC DE
OC59' C0 RET
OC5A' :
OC5A' : check '('
OC5A' :
OC5A' :
OC5A' ISI: LD A,(DE)
OC5B' FE 28 CP
OC5D' 20 RET NZ
OC5E' 13 INC DE
OC5F' C0 RET
OC60' :
OC60' : check whether line pointer is

```

リスト5 WZDソースリスト4

[illegible]

```

0129# 4F          CNAR:      LD      C,A
0131# 13          INC      DE
0132# 1A          LD      A,(DE)
0133# F2 80      CP      CR
0135# CA ***     JP      Z,ERR12
0136# FE 5E      CP      ****
0138# CC ***     CALL     Z,ESC
0139# 20 00      LD      H,0
0139# 0F        LD      L,A
0139# 13          INC      DE
0131# 1A          LD      A,(DE)
0132# 13          INC      DE
0133# 80          CP      C
0134# CD ***     JP      NZ,ERR12
0137# C9          RET
0138#
0139# 13 ***     EVAL20:   CALL     INC DE
0139# CD ***     CALL     SPCUT
0139# F2 26      CP      '('
0139# CA ***     JP      Z,PAREN
0141# FE 20      CP      '-'
0143# CD ***     JP      Z,EVINS
0144# FE 24      CP      '$'
0148# CA ***     JP      Z,EVHEX
0148# FE 25      CP      '*'
0149# CA ***     JP      Z,EVINL
0150# CD ***     CALL     ISDEC
0153# D2 ***     JP      NC,EYDEC1
0156# CD ***     CALL     ISO
0159# CA ***     JP      Z,EYCHAR
0159# 21 ***     LD      HL,ESPTRL
0159# 81 ***     LD      BC,SPREND-ESPTRL
0162# ED B1      CP      ZERR
0164# CA ***     JP      Z,EERRS
0167#
0167# C5          LABEL:   PUSH    BC
0180# CA ***     LD      A,(pass)
0180# A7          AND      A
0180# 28 20      JR      NZ,LABEL2
0187#
0187# 3A ***     LD      A,(UNDEF) :For Pass-1
0271# FE 62      OR      2      :A Label is used
0173# 32 ***     LD      (UNDEF),A
0176#
0176# C0 ***     CALL     ISEXT
0179# 26 18      JR      Z,LABEL5
0178# CD ***     CALL     SEALBL
0179# 38 04      JR      C,LABEL1
0180# FE 01      CP      1
0182# 28 29      JR      Z,LABEL4
0184#
0184# 32 83      LABEL:    LD      A,3      :Label is used
0180# CD ***     LD      (UNDEF),A      :A Label is used
0180# 18 32      LD      A,5      :Value is not certain
0180# 32 ***     LABEL:    LD      A,3
0180# 32 ***     LD      (UNDEF),A
0180# CD ***     CALL     SKRBL
0183# 13          INC      DE
0184# 13          INC      DE      :SKIP '8's'
0185# 18 26      JR      LABEL4
0197#
0197# 3A ***     LABEL2:   LD      A,(UNDEF) :For PASS-2
0194# FE 82      OR      DE
0196# 32 ***     LD      (UNDEF),A :A Label is used
0197#
0197# C0 ***     CALL     ISEXT
0242# 28 26      JR      Z,LABEL4
0194# CD ***     CALL     SEALBL
0194# FE 01      CP      C,ERR1 :Undefined Label Error
0194# 28 13      CP      1      :ASEG
0242# FE 85      CP      5      :EXT
0190# CA ***     JP      Z,LABEL9
0193#
0193# 32 83      LD      A,3
0195# 32 ***     LD      (UNDEF),A
0196# 1E 08      LD      A,08H
0194# CD ***     CALL     POSTABL
0190# 08          LD      H,B
0196# 08          LD      L,C
0197# C1          POP      BC
0198# C9          RET
01C1#
01C1# 50          LABEL:   LD      H,B
01C2# 09          LD      L,C
01C3# 32 E7      LD      A,#E7H
01C5# CD ***     CALL     PUTABL
01C9# C1          POP      BC
01C9# C9          RET
01CA#
01CA# CD ***     LABEL6:   CALL     LENLBL
01C0# 78          LD      A,B
01C0# FE 28      CP      21+1
0198# CD ***     JP      NC,ERR3 :Too Long Label Name
0193# FE 26      OR      0,28H
0198# 1A          CALL     PUT4
0198# 1A          LABEL7:  LD      A,(DE)
0198# CD ***     LD      A,ISALNUM
0198# 38 04      JR      C,LABEL9
019E# 13          INC      DE
0198# CD ***     CALL     PUT4
01E2# 18 F4      JR      LABEL7
0194# 32 83      LD      A,3
0196# 32 ***     LD      (UNDEF),A
019E# 13          INC      DE

```

リロケータブルアセンブラWZD 91


```

01521* CD 03 20 CALL DWTSTR
01522* D0 RET C
01523*
01524* LD BC,IMF_SIZE
01525* LD DE,(FCB_ADR)
01526* LD HL,FILE_BF
01527* LDIR
01528*
01529* RET
01530*
01531* RET
01532*
01533* FILE CLOSE
01534*
01535* LD (FCB_ADR),HL
01536* DE,FILE_BF
01537* BC,IMF_SIZE
01538* LDIR
01539*
01540* LD A,(FLOSK)
01541* LD (DSK),A
01542*
01543* LD HL,(FLSIZE)
01544* INC L
01545* DEC L
01546* NZ,WRITE
01547*
01548* LD A,(CRC)
01549* LD HL,(FLSIZE)
01550* INC L
01551* DEC L
01552* JR NZ,COL3
01553* INC A
01554* LD HL,A
01555* LD (FLSIZE),HL
01556*
01557* LD A.1
01558* LD DE,(DEBUF)
01559* LD HL,(_STBUF)
01560* CALL _DROSD
01561* RET C
01562*
01563* LD HL,FILE_BF
01564* LD DE,(FCB_ADR)
01565* LD BC,20H
01566* LDIR
01567*
01568* LD A.1
01569* LD DE,(DEBUF)
01570* LD HL,(_STBUF)
01571* CALL DWTSTR
01572* RET C
01573*
01574* CALL RDPTAT
01575* RET C
01576*
01577* LD B,10H
01578* LD HL,TBLCLST
01579* LD A,(HL)
01580* CP 7FH
01581* JP NC,WRAT
01582*
01583* INC HL
01584* LD C,(HL)
01585* PUSH HL
01586* LD HL,(_FATF)
01587* LD B.0
01588* LD E.A
01589* ADD HL,DE
01590* LD (HL),C
01591* POP HL
01592*
01593* DEC B
01594* JZ RDOPND ;Bad File Allocation
01595* CALL
01596*
01597* INPUT FROM FILE
01598*
01599* LD HL,(FVIV)
01600* ADD HL,HL
01601* DE,RDPTAT ;HL = RDPTAT + (FVIV) * 2
01602* ADD HL,DE ;INC (RDPTAT) lower byte
01603* INC HL
01604* PUSH HL
01605* LD BC,IMF_SIZE
01606* LD HL,(FVIV)
01607* CALL MUL
01608* LD W.0 ;W.0 = W.0 + IMF_SIZE * (FVIV)
01609* ADD HL,DE
01610*
01611* PUSH HL
01612* READ
01613* POP HL
01614* JR C,INP2
01615* LD DE,32H
01616* INC HL,DE ;HL points FILPNT
01617* INC HL
01618* POP HL
01619*
01620* LD A,(HL)
01621* INC HL
01622* LD A,(HL)
01623* OR A
01624* RET
01625*
01626* POP HL
01627* ;RET with CY = 1 (Error)
01628*
01629* DW RDOPND
01630* DW RDOPND + 100H
01631* DW RDOPND + 200H
01632* DW RDOPND + 300H
01633*
01634* LD BC,IMF_SIZE
01635* LD DE,(FCB_ADR)
01636* DE,FILE_BF
01637* LDIR
01638*
01639* LD A,(FLOSK)
01640* LD (DSK),A
01641*
01642* LD A,(FLPNT)
01643* LD B.A
01644* LD A,(CRC)
01645* CP B
01646* JP C,RDOPND ;Bad AllocationFile
01647*
01648* LD A,B
01649* LD HL,A
01650* LD HL,A
01651* LD HL,A
01652* LD HL,A
01653* LD HL,A
01654* LD HL,A
01655* LD HL,A
01656* LD HL,A
01657* LD HL,A
01658* LD HL,A
01659* LD HL,A
01660* LD HL,A
01661* LD HL,A
01662* LD HL,A
01663* LD HL,A
01664* LD HL,A
01665* LD HL,A
01666* LD HL,A
01667* LD HL,A
01668* LD HL,A
01669* LD HL,A
01670* LD HL,A
01671* LD HL,A
01672* LD HL,A
01673* LD HL,A
01674* LD HL,A
01675* LD HL,A
01676* LD HL,A
01677* LD HL,A
01678* LD HL,A
01679* LD HL,A
01680* LD HL,A
01681* LD HL,A
01682* LD HL,A
01683* LD HL,A
01684* LD HL,A
01685* LD HL,A
01686* LD HL,A
01687* LD HL,A
01688* LD HL,A
01689* LD HL,A
01690* LD HL,A
01691* LD HL,A
01692* LD HL,A
01693* LD HL,A
01694* LD HL,A
01695* LD HL,A
01696* LD HL,A
01697* LD HL,A
01698* LD HL,A
01699* LD HL,A
01700* LD HL,A
01701* LD HL,A
01702* LD HL,A
01703* LD HL,A
01704* LD HL,A
01705* LD HL,A
01706* LD HL,A
01707* LD HL,A
01708* LD HL,A
01709* LD HL,A
01710* LD HL,A
01711* LD HL,A
01712* LD HL,A
01713* LD HL,A
01714* LD HL,A
01715* LD HL,A
01716* LD HL,A
01717* LD HL,A
01718* LD HL,A
01719* LD HL,A
01720* LD HL,A
01721* LD HL,A
01722* LD HL,A
01723* LD HL,A
01724* LD HL,A
01725* LD HL,A
01726* LD HL,A
01727* LD HL,A
01728* LD HL,A
01729* LD HL,A
01730* LD HL,A
01731* LD HL,A
01732* LD HL,A
01733* LD HL,A
01734* LD HL,A
01735* LD HL,A
01736* LD HL,A
01737* LD HL,A
01738* LD HL,A
01739* LD HL,A
01740* LD HL,A
01741* LD HL,A
01742* LD HL,A
01743* LD HL,A
01744* LD HL,A
01745* LD HL,A
01746* LD HL,A
01747* LD HL,A
01748* LD HL,A
01749* LD HL,A
01750* LD HL,A
01751* LD HL,A
01752* LD HL,A
01753* LD HL,A
01754* LD HL,A
01755* LD HL,A
01756* LD HL,A
01757* LD HL,A
01758* LD HL,A
01759* LD HL,A
01760* LD HL,A
01761* LD HL,A
01762* LD HL,A
01763* LD HL,A
01764* LD HL,A
01765* LD HL,A
01766* LD HL,A
01767* LD HL,A
01768* LD HL,A
01769* LD HL,A
01770* LD HL,A
01771* LD HL,A
01772* LD HL,A
01773* LD HL,A
01774* LD HL,A
01775* LD HL,A
01776* LD HL,A
01777* LD HL,A
01778* LD HL,A
01779* LD HL,A
01780* LD HL,A
01781* LD HL,A
01782* LD HL,A
01783* LD HL,A
01784* LD HL,A
01785* LD HL,A
01786* LD HL,A
01787* LD HL,A
01788* LD HL,A
01789* LD HL,A
01790* LD HL,A
01791* LD HL,A
01792* LD HL,A
01793* LD HL,A
01794* LD HL,A
01795* LD HL,A
01796* LD HL,A
01797* LD HL,A
01798* LD HL,A
01799* LD HL,A
01800* LD HL,A
01801* LD HL,A
01802* LD HL,A
01803* LD HL,A
01804* LD HL,A
01805* LD HL,A
01806* LD HL,A
01807* LD HL,A
01808* LD HL,A
01809* LD HL,A
01810* LD HL,A
01811* LD HL,A
01812* LD HL,A
01813* LD HL,A
01814* LD HL,A
01815* LD HL,A
01816* LD HL,A
01817* LD HL,A
01818* LD HL,A
01819* LD HL,A
01820* LD HL,A
01821* LD HL,A
01822* LD HL,A
01823* LD HL,A
01824* LD HL,A
01825* LD HL,A
01826* LD HL,A
01827* LD HL,A
01828* LD HL,A
01829* LD HL,A
01830* LD HL,A
01831* LD HL,A
01832* LD HL,A
01833* LD HL,A
01834* LD HL,A
01835* LD HL,A
01836* LD HL,A
01837* LD HL,A
01838* LD HL,A
01839* LD HL,A
01840* LD HL,A
01841* LD HL,A
01842* LD HL,A
01843* LD HL,A
01844* LD HL,A
01845* LD HL,A
01846* LD HL,A
01847* LD HL,A
01848* LD HL,A
01849* LD HL,A
01850* LD HL,A
01851* LD HL,A
01852* LD HL,A
01853* LD HL,A
01854* LD HL,A
01855* LD HL,A
01856* LD HL,A
01857* LD HL,A
01858* LD HL,A
01859* LD HL,A
01860* LD HL,A
01861* LD HL,A
01862* LD HL,A
01863* LD HL,A
01864* LD HL,A
01865* LD HL,A
01866* LD HL,A
01867* LD HL,A
01868* LD HL,A
01869* LD HL,A
01870* LD HL,A
01871* LD HL,A
01872* LD HL,A
01873* LD HL,A
01874* LD HL,A
01875* LD HL,A
01876* LD HL,A
01877* LD HL,A
01878* LD HL,A
01879* LD HL,A
01880* LD HL,A
01881* LD HL,A
01882* LD HL,A
01883* LD HL,A
01884* LD HL,A
01885* LD HL,A
01886* LD HL,A
01887* LD HL,A
01888* LD HL,A
01889* LD HL,A
01890* LD HL,A
01891* LD HL,A
01892* LD HL,A
01893* LD HL,A
01894* LD HL,A
01895* LD HL,A
01896* LD HL,A
01897* LD HL,A
01898* LD HL,A
01899* LD HL,A
01900* LD HL,A
01901* LD HL,A
01902* LD HL,A
01903* LD HL,A
01904* LD HL,A
01905* LD HL,A
01906* LD HL,A
01907* LD HL,A
01908* LD HL,A
01909* LD HL,A
01910* LD HL,A
01911* LD HL,A
01912* LD HL,A
01913* LD HL,A
01914* LD HL,A
01915* LD HL,A
01916* LD HL,A
0191
```

94 Oh! X 1990.7.

復活のCGアニメーション

プロジェクトチーム DōGA かまた ゆたか

はじめに

2カ月連続して、連載を休んで申し訳ありません。“あまりの人気のなさに連載が打ち切られた?”“かまたが過労(あるいは急性胃潰瘍)で倒れた?”などいろいろ噂も飛び交い、当方にも何件かの問い合わせがありました。しかし実際には、特にこれといった理由はなく、文字どおり“申し訳がナイ”のです。今月はちょうど1周年ということもありますので、一部の連載陣も交代して、気分一新頑張っていきたいと思います。

さて、5月号に「アマチュアCGAコンテスト入選作品集」ビデオテープ配布のお知らせを掲載しましたが、皆さん申し込まれたでしょうか(すでに締め切りましたので、見落としていた方はご了承ください)。このビデオをご覧になれば、現時点でのアマチュアCGAのレベルというものがはっきりわかると思います。皆さんは、“この程度なら私のほうが上だ”と思いましたか? それとも“オオ! 私ニハ勝テナイ(出典:アジョージャ)”と諦めてしまいましたか? とりあえずCGAシステムで適当にカットを作れるようになってきた方でも、コンテストに応募するような“作品”という形にはまとめられないとか、ストーリーを思いつかないという方は多いのではないのでしょうか。

今年度のCGAコンテストの締め切りも、12月31日です。夏休みに本格的な作品制作を行うために、そろそろ作品構想に入らなければいけません。ということで、今回はCGAの作品企画を取り上げます。

作品企画

作品企画のメインは、もちろん制作する作品の内容を決めることです。そのほかに、制作のスケジュールも決めなければいけません。また、チームなどで共同制作する場合は各自の分担を決め、効率よく制作できるように準備します。作品の内容の決め方は、作品の種類や目的によってまったく異なり、画一的な手法があるわけではありませんので、のちほどいくつかのパターンをまとめて紹介します。

それではまず、CGA作品企画において最も大切なポイントについてお話ししましょう。それはズバリ“作品を完成させる”ことです。これを常に最優先にすることを

今回は、そろそろ本格的な作品制作に取り掛かろうという方のために、CGA作品の企画について考えてみたいと思います。また、めったに連載1周年を記念して、バージョンアップサービスのお知らせも掲載いたします。

念頭においてください。作品が完成しなければ、それまで行ったすべての努力とアイデアが水の泡と化してしまうのです。

“作品を完成させる”企画を立てるためには、自分の持っているパワーを正しく認識する必要があります。自分の持っているパワーというのは、単に“努力と根性”ということではなく、技術的なパワー、CGA制作に割ける時間的問題、自分の持っているハード上の制約、制作に参加する人数と各自のパワーなども含みます。誰しも作品企画の段階では、いままでになかったような凄い作品を作ってやろうと思うものです。そういった意気込みは大変大切なもので、そのくらいでなければまともな作品にはならないのです。が、凄い作品には凄いパワーが必要になるので、結局はオーバーワークになり、“未完の大作”となってしまうがちです。今回制作するのは、凄い作品を作るための習作なのだと自分にいい聞かせて、“ちょっと努力すれば、すぐできてしまう”と思える程度の作品を企画するのがコツでしょう。

作品を完成させるためにもうひとつ考えなければいけないのは、目的をはっきりさせることです。学園祭で上映するための作品なのか、新入生募集のPR用なのか。別に期限などない気ままな作品というものもあるでしょうが、期限のない作品は完成した試しがありません。いつまでに必ず完成しなければいけないという必然性を持つことは非常に大切です。最初はすばらしいと思った企画でも、制作の過程では、なかなか作業がはかどらなかつたり、思ったようなイメージが表現できなかつたりして、だんだん嫌になってくる時期がきます。そんな時期を乗り越えて完成させてこそ、作品制作の喜びが体験できるわけですから、途中でくじけないために、この必然性を企画の段階から考えてください。

特に目的がない方は、当方が主催する「アマチュアCGAコンテスト」での入選を目標に励んでください。そうすれば、期限は自動的に今年の年末になります。そのほか、友達の前で、“何月何日までにこの作品が完成しなければ、私のX68000をDōGAにカンパしてやる”と宣言し、自らを窮地に追い込むのもよいでしょう。でもこの場合、当方としては完成しないほうがうれしく思います。

次にスケジュールの組み方ですが、これは目的(期限)と自分のパワーがわかればだいたい決まってしまうと思います。要は、スケジュールなんて所詮遅れるものなのだから、真剣に期限までに完成させる気があるのなら、余裕を持

って計画すればよいのです。以下に、サンプルを挙げますので参考にしてください。

条件1: CGAコンテストに出品する

条件2: 7月後半から9月初頭は夏休みで余裕がある

条件3: 9月後半から10月初頭は試験で時間がない

企画(絵コンテ含む)	6月下旬~7月下旬
モデリング	7月上旬~7月下旬
モーショングデザイン	7月下旬~8月下旬
レンダリング	8月上旬~9月中旬
編集/仕上げ	10月中旬~11月初頭
試写会	11月初頭
改良	11月上旬~11月下旬

企画からレンダリングまでの作業は、お互いにオーバーラップしていますが、年末に完成するのなら、8月に入る前にすべての形状デザインを、9月に入る前にモーショングデザインを終えておくべきだと解釈してください。この例では、余裕を持たせて、コンテストの締切のひと月前に完成することになっていますが、その程度の余裕では、締切の前夜は徹夜することを覚悟したほうがよいでしょう。11月初頭の試写会というのは、大阪大学の学祭のことですが、コンテストで本気に入賞を狙うのなら、ひと月前には最初のバージョンを完成させておいて、ほかの人の意見を聞いてみることをお勧めします。試写会の場がなければ、当チームにお送りくださいますと、寺田からの“教育的指導”があるでしょう。

作品企画例

1) かおる流 明日は明日の風が吹く法

数年前に、かおる君が制作した「ずんぐり&むっくり」は、エンタープライズもどきの“ずんぐり号”とリリアントもどきの“むっくり号”との戦いをコミカルに、メルヘンチックに描いた心温まる(?)作品です。宇宙空間をすっとなばしていたずんぐり号の前に、むっくり号が現れ、びっくりしたずんぐり号は急停止します。続いて、むっくり号はフェイザーを発射してきたので、ずんぐり号も光子魚雷をお見舞いしてやりました。光子魚雷を受けたむっくり号は、小さな3つのむっくり号に分裂してしまいます。そこにずんぐり号が近寄って、放り投げたり、背中に乗せたりして遊びます。再び3つのむっくり号は合体して、今度はずんぐり号と同じ姿になってしまい、2台仲よく宇宙の彼方へ飛んでいきます。

この想像を絶するわけのわからないストーリー展開はどのようにして生まれたのでしょうか。実は、この作品においては企画なんてものは存在しません。とりあえず、1カットを作り、それができたあとで、次の1カットをどのようにするかを考えるという、実にいいかげんな方法で制作されています。初心者なのに、何月何日までにとりあえず何か見せるものを作らないといけないという方にはよい方法といえます。また、どうしても作品を思いつかないときの最後の手段としても有効でしょう。コツは、とにかく次の1カットのことしか考えない。次の1カットをいかに面白く、あるいは意外性をもたせるかだけを考えて、どんどんつなげていくのです。そして、

ある程度の量ができたら、あるいは締切が近づいてきたところで、強引にオチをつけて、終わりにしてしまえばよいのです。この方法のよいところは、誰でもすぐに作品制作に掛かれるということと、一応ひとつの作品を完成することができるという点です。ただし、かなり偶然性に頼っているわけで、傑作ができる可能性はあまり多くないかもしれません(作者のセンス次第ですが)。

2) げんし流 ばちもん法

「Mの喜劇」という、げんし君が制作したこれまた古い作品があります。これは、アカデミー短編賞にもノミネートされたかの名作、PIXAR社の「Luxo Jr.」のパロディです。非常に有名な作品なのでご存じの方も多いと思いますが、原作では電気スタンドの親子とゴムまりが出てくるところを、「Mの喜劇」ではパソコンの親子とピラミッドに変えています。やはり、げんし君が「Luxo Jr.」の大ファンというのが制作の動機のようなのです。

また、チームTOSAKAがCGAシステムで制作した「スタートレック・カーンの圧勝(CGAコンテストのオープニング)」は、映画版スタートレックのパロディです。さらに、作者は存じませんがガンダムのキャラクターがSTAR WARSのデススターに突入するという「ハロZZ」という作品も見つかることがあります。

これらの作品は、自分にとって特別思い入れのある既存の作品を自分の手でCG化してしまうというパターンで、異様なパワーを発揮し、技術的にも完成度も高いという共通点があるようです。その異様なパワーの源がなんにしろ、自分の作品に対してのめり込めるというのは大変よいことで、この手の作品の大きな強みです。

ただ、パロディである分だけオリジナリティは少なく、

初めて読む読者へ

この連載では、D6GA・CGAシステムを中心に、CGアニメーション作品の作り方を取り上げています。D6GAとは何か? CGAシステムとは何か? 簡単におさらいしておきましょう。

プロジェクトチームD6GAとは、“手軽でパーソナルな映像としてのCGアニメーションの普及”を目的に、大阪大学コンピュータクラブや京都大学マイコンクラブが中心になって活動しているアマチュアの団体です。そしてCGAシステムとは、D6GAによって開発された、パソコン(X68000)上で手軽にCGA作品制作を行うためのプログラムです。このCGAシステムを、私たちの活動に賛同するアマチュアに限って、実費+カンパで配布すると本誌上でも発表したところ全国から異様にたくさんの申し込みをいただき、アマチュアCGAの活動は一気に全国に広がりました。そこで、当チームの活動もシステム開発だけでなく、CGA作品制作、上映会、アマチュアCGAコンテストの開催、全国のアマチュアCGA団体との連携など幅広い活動を展開しているのです。

なお、本連載の経過は以下のとおりです。入手ご希望の方は、本誌バックナンバー案内をご覧ください。

●1989年7月号 ついに完成! D6GA・CGAシステム: D6GAプロジェクトとCGAシステムの概要、および申し込み方法

●8月号 CGA初心者救助隊出動!: まったくの初心者でもできるCGAシステムの使い方

●9月号 宇宙要塞CADを攻略せよ!: CADによる3Dの形状デザインのテクニック

●10月号 パンドラの箱が開くとき: CGAシステムの隠し機能の公開

●11月号 いぶし銀はどんな色?: アトリビュート(色、材質のデータ)の設定の仕方

●12月号 くさってもFFE: モーションエディタFFEの使い方

●1990年1月号 脳ミソどろどろフレームソース: CGA共通規格による動きの記述方法

●2月号 1990年運営基本方針発表!: ロボットなど構造体の記述と全国各地のチーム募集

●3月号 映像表現のテクニック: 映像作品としてのCGAを仕上げるための入門

●4月号 春だ、4月だ、新歓だ!: 2月号で募集した各地のCGAチームの紹介

コンテストを目的とした場合は多少不利になるのはしかたないでしょう。しかし、逆にいえば新歓のPRなどが目的の作品ならば、全然気にする必要がないということですよ。また、パロディでもオチを変えたりしてオリジナリティを出すということは十分可能だと思います。

この手の作品を制作すると、原作に近づけようと努力する過程で、一流作品の“見せ方”や“編集”のテクニックを知らず知らずのうちに身につけるといって、大きなメリットがあります。自分の手でCG化してみたい映像に心当たりのある方は、一度トライしてみるのもよい勉強となるでしょう。

3) 雪だるま流 BGM法

「冬の終わる夜」は、全日本ビデオコンテストへの出品と、女性へのCGAのPRという打算に満ちた意図で制作されました。まず最初に夢で見たイメージがあり、それを音楽担当者に伝えて作曲してもらいました。具体的な内容はすべてこの音楽に合わせて考えていきます。

この、BGMを先に決定してそれに合わせて映像を企画するという手法は、非常に汎用性があり有効です。音楽によってイメージがはっきりしているの、ストーリーや映像を想像しやすく、自然と全体がまとまります。極端な話、脈絡のないカットのつなぎ合わせのようにストーリー性のないものでも、音楽に合っていれば、ひとつ

の作品になるといってもウソではありません。センスがよければ、観客に感動を与える名作にもなり得ます。さらに、通常のメディアと違ってCGAは、コマ1秒単位の制御をデジタルに処理できるため、音楽にぴったり合わせるという作業も比較的楽です。

問題点としては、まず著作権問題が挙げられます。個人として楽しむならいざ知らず、コンテストに出品する場合主催者側に曲の変更を求められる場合もあり、作品の魅力が半減します。アマチュアCGAコンテストでも、著作権問題がクリアされていない作品は基本にお断りです。この手の作品の場合、音楽は最も重要な要素になるので、オリジナル曲で頑張ってもらいたいと思います。しかし、オリジナル曲ができないのであれば、BGM用の著作権が放棄された曲(レンタルレコード店にあるそうです)を利用するか、ちゃんと著作権協会に許可を取るようにしましょう。もうひとつの問題は、作品の長さなどが強制的に曲に縛られるという点です。音楽をMIDIデータとして持ち演奏させるなら、映像の都合に合わせて一部を編集することも可能となります。また、曲をフェードアウトで強引に終わらせてもよいでしょう。

この手法の場合、当然企画は音楽を決めることから始まります。それはもう、かたっぱしから音楽を聴くしかありません。自分の好き嫌いよりも、映像を思い浮か

各読者通達事項

バージョンアップサービスのお知らせ

連載1周年を記念して、CGAシステムのバージョンアップサービスを行います。新しいバージョンは「2.11」あるいは「2.12」です。「2.0*」との主な違いは下記のとおりです。あまり根本的な変更はないのですが、なかには役に立つ機能やプログラムもありますので、ご希望の方は申し込んでください。

今回のサービスの内容は、マニュアルなどもなく、ディスクに変更のあったプログラムだけを入れて送るという簡単なものですので、実費は500円程度ですむと思われます。前々からの約束どおり、バージョン「2.00」および「2.01」をお持ちの方には基本的に無料(カンパはご自由)とさせていただきます。

なお、サービスのスケジュールは以下のようになる予定です。

申し込み開始	6月18日
申し込み締切	7月31日
発送開始	8月下旬
実費等の払い込み	受け取り後2週間以内 (用紙は同封します)
苦情(未着等)対応	9月中旬～9月30日

●バージョンアップの申し込み方法

必ず官製ハガキに、次の点を明記して当プロジェクトルームまでお送りください。記入ミスなどによるトラブルには対応致しかねますし、「マニュアルを送れ」「2セット送れ」「コンテストのビデオまだありますか?」といった、別の用件をいっしょに書いてこられても、個別対応はできませんのでご了承ください。

- 1) 氏名
- 2) 住所(郵便番号)

3) 電話番号

4) お手持ちのCGAシステムのバージョン

5) 下記のアンケート(複数回答可)

アンケート

このCGアニメーション講座の連載において、どのような内容を望みますか?

- a) マニュアルの代わりとなるようなCGAシステムの使い方
- b) マニュアルを読んでもわからない人のための入門講座
- c) マニュアルに載っていないような高度なテクニック
- d) CGA作品制作法
- e) D6GAにおける作品制作の実況レポート
- f) 映像制作一般論
- g) CGのアルゴリズムやプログラム開発方法
- h) イベントレポートなどのD6GAの活動報告
- i) その他(ご自由に記入してください)

●主なバージョンアップの内容

・REND、FFE、ATRの各プログラムがバージョンアップした。

REND: バックグラウンド機能、若干の高速化、背景呼び込み、空気遠近法

FFE: ロード機能、若干の高速化

ATR: 材質感表示が可能、操作性を一新
・Ver2.0* を発表してから新しく作られたプログラム(HANIM、IC、CRD、SMOOTH、SUF CUT、STAR、BOMB、PATIPIC)が入った。

HANIM: SRANIMを2倍高速化(ただし65536色中256色)

IC: 512の画像データの一部(256)を連続的に切り出す

CRD: 65536色の画像データを256色に落とす

SMOOTH: テロップ画像データのアンチエイリアス化

SUF CUT: 形状データの一部を切り出し、部品化する

STAR: 背景用の星空を生成する

BOMB: 形状データを爆発させる

PATIPIC: Z'sSTAFFに常駐し、CGAシステム用の画像ファイルを出力する

・CADが新しいIOCS(SX-WINDOWについているもの)に対応した。

・HumanのシステムがVer2.01になった。

なお、Ver2.10とはいくつかのバグがとれた程度で、基本的には変わっていません。

▶ X68000最新機種をお持ちの方へ

PROII、EXPERTII、SUPER-HDなどの最新機種では、IOCSが変更されており、CGAシステムの一部のプログラムを使用したとき不都合が出るのがわかりました。また、旧機種に最新の「IOCS.X」を組み込んだ場合も同様です。とはいっても、マウスカーソルのあとが残るときがあるという程度のもので、まったく使えなくなるわけではありませんし、若干高速化されるので、私は気にせずに使っています。

今回のバージョンアップで対応しようと思っていたのですが、問題のあるすべてのプログラムを全部直すのは間に合わないと思います。一応の対応策を記しますので、ご注意ください。

- ・気にしない
- ・CGAシステムディスクから起動する
- ・CONFIG.SYSの中でIOCS.Xを組み込むのをやめる

べやすいかどうかという基準に重点をおきます。また、イメージがひんぱんに変わってくるような曲も、映像をぴったり合わせるのが難しいのでやめておいたほうがよいでしょう。それから、歌詞にはあまり捕らわれないでよいと思います。外国の歌詞なら、どうせ誰もわからないし、日本語の歌詞でどうしてもじやまになるようでしたら、カラオケを利用するのもひとつの手です。チームで制作するときは、ほかのスタッフに話す前に曲を決めておくほうがよいでしょう。曲によってイメージを共有できるのはこの手法の利点といえます。

曲を決めたら、何度も何度も聴き、映像のイメージを固めていきます。最終的には、その曲を聴けば頭の中で全カットの映像が浮かぶまで詰めます。そして、楽譜を見ながら、“この音が入る瞬間にカメラはここを向いているから……”という具合に絵コンテを制作すると、フレーム数まではっきりするコンテができるでしょう。

4) 令子流 4 カット法

これは前々から一度チャレンジしようと思っていたトッテオキの企画ですが、4コマ漫画と同じように、起・承・転・結をそれぞれ1カットで表現し、4カットのみでひとつの作品を構成する手法です。ですから1カットが5秒程度とすると、タイトルを入れても30秒足らずの小作品ということになります。短いだけあって、制作は簡単ですし、起承転結の構成を身につける練習にもなります。いくらなんでも30秒というのは短すぎるというのでしたら、同じ主人公キャラを使用した作品を毎月1作ずつ制作し、年末には6本まとめたオムニバスという形式を取ればよいのです。そうすると、途中で急用ができて、制作が1、2カ月遅れても、本数は少なくなってもちゃんと作品は完成するというメリットもあります。それに、新入部員に対する練習課題にも適しています。

問題点は当チームでも試してみたことがないからわかりません。やはりオチが決まらないと見るに耐えないので、アイデアが勝負でしょう。どうしても、ギャグ、コメディが中心になると思いますが、まったく新しいジャンル

を開拓する余地も多いと思います。

5) VOYAGER流 点と線法

昨年制作した「Thank you VOYAGER」では、海王星の影でVOYAGERが奮闘する様子を描くことで宇宙開発に携わっている方々への尊敬と感謝の念を表現しようと試みました。まず最初にイメージしたのが、“ゆっくりと回転する海王星に吸い込まれていくVOYAGER”のカットと、“VOYAGERから送られてくる分析図に宇宙開発史がだぶってスクロールする”といったカットです。次に“太陽系を離脱し、暗い宇宙空間に消えていく”

“BGMがフェードアウトし、無音で暗黒の影の部分に入っていく”“逆光を浴びながら影から出てくる”カットなどを考えました。これらは、すべて連続性のないバラバラのカットであり、ストーリーとしてつながっているわけではありません。すなわちこれが“点”つまり“見せ場”なのです。そして、“点”をたくさん考え、“線”でつなぐことでストーリーを作るのがこの“点と線”法です。

もう少し詳しく解説しましょう。まず、“点”を思い浮かべたら、どんどん紙に描いていきます。必ず、ひとつの“点”につき、1枚の紙を用い、非常にラフなイラストと解説を添えます。この段階では、ストーリーとか、構成とかいうものを考える必要はなく、関連ありそうなカットやアイデアを蓄えます。約5分の「VOYAGER」で、1週間以上をこの作業に費やしました。次にそれらの紙を床にばらまき、スタッフとともにいろいろ並べてみましょう。“これとこれは一連のシーンとしてつながるのでまとめておこう”とか、“このカットは全然別なのでこっちの隅に置いておこう”とか、“これはエンディングに使えるから右端だ”とか分類できてきます。そしてその作業が進むと、だんだんいく通りかのストーリーが見えてくるでしょう。“このへんのとこのへんのを使って、このエンディングにもっていけば、3分ぐらいの短編ができそうだ”とか、“その順番を入れ替えて、その前にこの辺のをくっつけると話が膨らんで5分ぐらいの作品になる”という具合です。基本的には、できるだけ多くの

モデラー高津のLOGIN

突然ですが、松井さんが忙しくなったため、SIGOPを引退することになりました。かわりに5月からはこの私、モデラー高津がSIGOPをしています（といっても私が暇というわけではケシテありません）。

ということで、その副作用により、今回からこのコーナーも“モデラー高津のLOGIN”となることになりました。私は知人ぞ知るあの“パロレイバー”の作者です。D6GAにおいても、あるときはプログラミング、あるときはモデリング、月に2回「花とゆめ」を持ち込むなんてこともしています。苦情係も兼任しているので、文句のある方はどうぞ。私は誰の挑戦も受けません。

今後のネットでの予定ですが、従来どおり、ユーザーの質問に対するQ&A、バグレポート、新しいツールのアップはもちろん行っていきます。さらに、ユーザー同士の交流の場として活用してもらいたいと思っています。最近、名古屋

のチーム“鮮色映人”からアルファベットの形状データがアップされたり、ネットもなかなか活発です。このようなユーザーからのアプローチは、SIGの本来の目的であり、たいへん喜ばしいことですので、データのアップもどんどん増えてほしいものです。

ところで、最近TVアニメのロボットの形状データを作ったというお便りをよくいただきますので、著作権問題について専門家のご意見を伺ってみました。

D6GA「もしもし、日本サンライズですか？少々お伺いしますが、御社のTVアニメの形状データをネットにアップすると罰せられるでしょうか？」

日本サンライズ「厳密に言えば問題がないわけでは 아니 ませんが、そのデータを有料で販売するなどしなければ、別にかまいません」

……ということですので、みなさんかっこいいデータを作ってアップしてください。

今月のアップデート

今月は半端なものではありません。チームTOSAKA制作、CGAコンテストオープニングアニメーションに使用された「エンタープライズ&リライアント」です。これを語るのには「努力と根性」という言葉しかありません。ネットにアップするときは、OPアニメーションのワンカットのフレームソースと一緒に載せるつもりなので、楽しみにしてください。

*

CGAシステムに関して質問がございましたらD6GAプロジェクトルームの「当てにならないアフターサービス係」宛か、次の各ネットにメールで送ってください。それでは、またD6GA・CGA NETでお会いしましょう。

ハンドルネーム

Taka2	J&PHOTLINE	JH082274
アスキーネット	PCSPCS33268	
PC-VAN	VAM28517	

“点”を通るような“線”を作ればよいのです。が、無理に全部“線”上に乗せてねじ曲がったストーリーになってしまうのは意味がないので、せっかくのアイデアとはいえ、次回に利用しようと潔く諦めてください。

大筋が見えてきた時点でこの作業は終了し、もう一度“点”を作る作業に戻ります。なぜなら、このままでは“点”が密集した部分と、直接つながらないのを“線”で強引につなげただけのところの差が激しいからです。本当に“線”でつながれるのかをもう一度チェックするとともに、その線上にくる“点”を見つけていかなければいけません。つなぎのカットばかりが連続するシーンなんて、中だるみそのものです。

この“点と線”法は、ある意味で作品企画としていちばんオーソドックスな手法といえます。また、練習すればすべての作業を頭の中でできるようになるでしょう。

実録! 作品企画

具体的に作品企画はどのように決定されるのか。4月21、22日にDōGAにて行われた作品企画会議のようすを紹介しますので、参考にしてください。

21日は、大阪大学コンピュータクラブの入部説明会があり、特にCGに興味のある新入部員はその日からいきなりDōGAのプロジェクトルームに連れてこられる。今回の企画会議は、この新入生にゴールデンウィークに制作してもらう作品を決めるもので、期間が短いこともあり“簡単にできる”という条件が初めから与えられていた。

まず最初に、作品制作のスタッフを決めなければいけない。“いきなり作品制作にチャレンジする者は誰かいないか”という問いに、いち早く手を挙げた3人組（以後、新米A、B、Cと呼ぶ）がスタッフに決定した。また、

中学時代から作曲、演奏などをやっており、音楽だけではなくプロモーションビデオなどの映像も手掛けてみたいとやってきた新人M（Mは“MIDI”の“M”）に、一応の監督をしてもらうことにした。さらにこの4人の新入生をサポートする上級生として、寺田、MAX田口、モデラー高津、本連載では出番のなかった砂川が参加し、私もプロデューサーとしてまとめ役をすることになった。

21日、新入生には、当チームで制作された作品やアマチュアCGAコンテストの作品を見てもらい、アマチュアCGAの現状やレベルを把握してもらった。また、CGAシステムの実演を交えながら、CGA作品制作の流れと、CGAシステムには表現しやすいものとそうではないものがあることを解説した。とりあえず新米3人組には、明日までに作品のアイデアを考えてくるようにいって、解散し、Mにいろいろ話を聞いてみることにした。監督とプロデューサーとの意思疎通は大切なのだ。

Mが目指している映像は、音楽と同じように、テーマなどというものはなくても、カッコいいとか、面白いといった、感覚的なフィーリングやイメージで作るべきだという。私はそれには反対で、映像がコミュニケーションの手段である以上、制作者の意図やテーマというのは作品制作において重要な意味を持つと思う。

このように、作品企画の段階でスタッフ内の意見が食い違うことはよくある。また制作の過程においても、このカットは省略するかどうか、色を変えてレンダリングし直すかどうかなど、もめごとはいくらでも起こり得る。そんなときは、迷わず監督（あるいはリーダー）に一任しよう。そして、どうしても監督の指示に納得がいかない場合でも、とりあえずその作品を完成させるために協力する。そしてそのあとで、今度は自分が監督になって作品を作ればよいのである。もちろんそのときは、前の

全国アマチュアCG団体一覧 追加版

前回の掲載に間に合わなかった方々です。参加希望の方は、下記の連絡先に自己紹介を送ってください。詳しくは4月号をご覧ください。

① チーム名 ② 代表者名 ③ 連絡先 ④ 活動内容 ⑤ 入会制限 ⑥ 構成人数 ⑦ その他

*

- ① 011 DōGA
- ② 清水拓詞（しみずひろし）
- ③ 〒059-19北海道勇払郡追分町本町6丁目39番地
- ④ 情報公開、CGA研究会、交歓会、その他……
- ⑤ DōGA・CGAに関心のある、やる気のある方。男女、年齢問いません。
- ⑥ 4人
- ⑦（現在の問題点より）CGAに関する資料不足、人材不足。何をしようか、どんなことができるかなどいろいろ考えてみましたが、何にも決まってません。私は、プログラムなどBASICがちょっとできるくらいです。ちなみに私は、学

生です。ヒマなのは私だけなのです。だからこの先どうなるかなんぞ、わかったもんじゃありません。できる限り頑張っていきますが、責任は持てませんのでそのへんをよろしく

- ① レッド・スターレッツ
- ② 松田英明
- ③ 〒279千葉県浦安市堀江4-9-40モア415
- ④ CGを用いた動画を作る（実はよく決まっていなかったりする）
- ⑤ 暗いやつはだめ。女の子は水着審査並びに写真審査の上……ジョークです。アブナくなければOK!
- ⑥ 2人
- ⑦ いかんせん社会人なので暇がない。人がいない! 理屈なんて知らなくていいよっていつてるのに。シナリオ作人とかデザインする人とかでもよいと思うんですけどね
- ① Ari Assembly
- ② 本橋正成（もとはしまさなり）
- ③ 〒153東京都目黒区祐天寺1-23-21
- ④ 代表者が素人なので助け合っていきたい。月に1回ぐらゐの茶会とIT-VI200を使ったNETを開く予定である。初心者、女性の方大歓迎! 当然CGA、プログラム、音楽を中心に真面目に遊んでしまう
- ⑤ 特になし。気に入ったら62円切手同封の上、

自己紹介を書いて郵送していただきたい。モデルがあるとう利だ!

- ⑥ 新設団体
- ⑦ 1. 代表者が素人（ただいま勉強中）2. 2月現在、X68000ACE-HDのHDがクラッシュ!（修理中）3. ネットを開く時間以外のtelは私がとってしまう（研究中）4. 金がない……
- ① Tak-CGAチーム（仮称）
- ② 清野文雄
- ③ 〒957新潟県新発田市古寺146
- ④ 情報交換。および作品制作の作業分担など
- ⑤ いまのところ、アクティブに情報を交換したいので、Tak-NET(BBS)会員の希望者で始めます（そのうち軌道にのってたらオープン的にやりたい）
- ⑥ 新設団体
- ① GINAX
- ② 三品徹（みしなとおる）
- ③ 〒602京都市上京区御西裏上ノ下立売上ル北町570-1
- ④ 作品交換など
- ⑤ 京都市に住んでいる人
- ⑥ 2人
- ⑦ いままで音楽の活動はやってたが、CGAはやっていない
- ① まだ考えていません（人が集まりしだい考

監督も1スタッフとして参加すべきだ。

今回の場合、とにかくMの好きにやらせることにした。
“何がかっこいいか”ということもテーマになり得るし、もしMが本当にただかっこいいだけの作品を作り続けるなら、いずれ物足りないと感じる日がくるだろう。

翌日、Mが最初に現れたので、2人でもう少し具体的な内容を詰めてみる。本人の希望も考慮して、Mのオリジナル曲に合わせたイメージビデオみたいなものを制作してみないかと提案する。Mも異存がないという。各自が独自に作ったカットを集めて適当に編集するというやり方で、とりあえず作ってみようということになった。この方法だと、期間が限られている場合でもできあがったところまでで作品にすることができるとは、厳密な打ち合わせや準備が必要なく、個々に独立制作できるメリットがある。しかし、画像としては脈絡がなく、作品としてのまとまりに欠けるのが多少心配だ。

独自に制作するといっても、出てくる物体だけでも統一しておくべきだろう。私は、「トロン」のような電子バイク、「ウイニングラン」のようなカーレース、「STAR WARS」のような宇宙戦闘もの、「グラディウス」のようなシューティングゲームの3D化という4つのパターンを考えた。特に最後のシューティングゲーム案は、敵キャラの形状デザインが楽しいし、最後にボスキャラを倒せば形だけでもオチがつき、まとまりがよいので美味しいと薦めてみた。が、Mは自分の曲のイメージからすると、電子バイクがよいというのでそれに決まった。

ちょうどMが自分の曲のテープを持ってきているというので、聞いてみることにした。確かにテンポがよい曲だ。が、はっきりしたメロディがなく、ゲームのBGMか間奏曲といった感じだ。脈絡のない映像と、メロディのない音楽では、映像と音楽どちらもメインになり難しく、

ピントのばやけた作品になってしまう。多くの者から同様の指摘を受け、さすがにMも認めざるを得なかったが、ここはひとつなんとかMに頑張ってもらうことにした。

そのころになって、やっと新米3人組がやってきた。それぞれちゃんとアイデアを考えてきたというので、みんな期待して集まってきた。「あんまり寝ぼけたことという、物が飛んでくるから気をつけや」と脅かす先輩たち。本当に投げつける物を手にする者までいる。

新米A「私は3つ考えてきました」

先輩「ひとりで3つとは偉いやんけ（別にヤクザではない。大阪弁である）」

新米A「1つめのタイトルは“地上げ屋”」

先輩「なんやねん。わけわからんがなあ。まあストーリーを聞かせてもらおう」

新米A「貧しい家族が食事をしているところに、地上げ屋の団体が乗り込んできます」

先輩「おいおい、それはCGか？ 実写か？」

新米A「CGです」

先輩「貧しいという雰囲気を出すのも難しいが、CGで自然な人体を描くのはちょっと無理やで。“日本昔ばなし”よりもずっとデフォルメして、描くとまんがチックになるが、この話はコメディなのか？」

新米A「シリアスです。地上げ屋は、いろいろ嫌がらせをするんですが、お父さんが頑張って、みんな追い払ってしまいます」

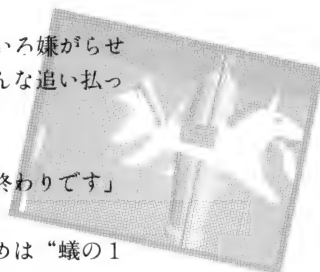
先輩「それから？」

新米A「翌朝、戦車がやってきて、それで終わりです」

この瞬間、いろんな物が投げつけられた。

新米A「まあ次の聞いてください。2つめは“蟻の1日”です」

先輩「今度はまともそうな」



える)

② 藤井信行

③ 〒640-04和歌山県海南市高津667

④ 参加者が何人かあれば（人数によって）活動内容が変わってくるので、人数がそろいしだい内容を考えていきたいと思います

⑤ アマチュアの人。CGAに興味のある方で熱心な人。なるべく高校生以上の方歓迎します。できれば和歌山の人。大阪の人でも可能

⑥ 約5名

⑦ 活動歴は2年ぐらいで、まだ活動というほどのことはやっていないが、C言語を2人で勉強している最中です。あとひとりC言語を使いこなしているようです。けっこう頻繁に活動しています。しかし、X68000を使いこなせる人が少なく、ゲーム以外のソフトが高額で手に入りにくいこと、人数の少ないこと、パソコンフレンドが近くにいないことなど問題点は多い

① L.M.D. (L.M.D. is a member of the team)

② 平田剛

③ 〒761-07香川県木田郡三木町池戸769報徳苑103号

④ D6GA・CGAシステムによるCGAの制作。

⑤ 香川医科大学まで来ていただける方で、高校または大学に在学中の方。コンピュータに関する知識のない方でも結構です（HD付きの

X68000またはPC-98をお持ちの方、大歓迎です）

⑥ 2名

⑦ 1988年から活動を開始し、今年で3年目を迎えます。細々と活動を続け、今年のアマチュアCGAコンテストに初めての作品「Memory」を出品しました

① “彩色映人”（さいろえいと）

② 上田晃好（うえだてるよし：RANA）

③ 〒457愛知県名古屋市中区西又兵衛町1-9・アクセスダイアル 052(614)0503（ネット専用24時間）

・プロトコル N81XN 2400MNP5
ID=“GUEST”（パスワードなし）

④ 名古屋地区を中心に、D6GAチーム専用連絡ネットを開設しています。D6GAからのデータなどを流すだけでなく、独自の企画を作り、プログラムやデータをみんなで楽しみながら共同制作していきたいと思っています

⑤ できればパソコン通信をやっていたりしゃる方、あるいはやろうという方

⑥ 約4名

⑦ ただいま、タイトル文字の形状データとそのコンバータに取り組んでいます。また、CGAシステムの勉強会、コンテストのビデオや「3・6・5」などの上映会と配布も行っています。

① 電脳クラブ電脳隊

② 辻村高

③ 〒810福岡市博多区博多駅前1-4-1シャープOAショールーム内「電脳クラブ」

④ 面白ければなんでも……

⑤ 老若男女問わず

⑥ 不定

⑦ 代表者自己紹介：九州地区のチーム紹介に、福岡がなかったのを見て思わず立候補しました。皆さんの積極的な参加を待ちしています。近々クラブで主宰しているBBSにD6GAのボードを開設しようと思います

博多っこBBS 092-481-3520

① コミュニケーション・アーツ68(CA68)

② 安藤優子

③ 〒813福岡市東区香住ヶ丘2-14-25エスポアル102号

④ 主にビジュアル制作。人に訴えかけ、なんらかの形でコミュニケーションしてゆくことが目標です。2D、3D、音楽などいろいろな手段を活用したいです

⑤ 時間さえあれば、連日連夜の徹夜で無償のチーム活動に積極参加できる人

⑥ 4名

⑦ 問題点：人手不足、資金不足、スイミン不足。「テクニク先行」より「全体の流れ」を重視したアニメーションを作成中

新米A「地下にアリの巣があって、女王アリの部屋やら、食料庫やら、たくさんの小部屋に分かれています」

先輩「なかなかCG化しやすくていいじゃないか。それから？」

新米A「そこまでしか考えていません」

再び、物が飛んだ。

先輩「それで、3つめはなんや」

新米A「3つめは“居酒屋の夜8時”」

先輩たちは何もいわず、投げる物を用意する。

新米A「ある男の視点から見た映像から始まります。夜の町を歩いていると、暗い通りに1軒の居酒屋から光が洩れています」

先輩「映像的イメージがしっかりしているのがいいな」

新米A「そのまま入って行くと、おかみさんが“今晚は遅かったのね”と声を掛ける」

先輩「何度もいうけど、雰囲気が出るほどの自然な人体は難しいで。まあいい、それから？」

新米A「そこまでしか考えていません」

3度、物が舞う。

先輩に物を投げつけられても最後までいった新米Aは、なかなか根性がある。将来、DōGAを背負う大物になるだろう。初心者なのだから、マトはズレ的なアイデアになってしまうのは仕方ないだろうし、新米Aのいうようなことが簡単にできない現状のほうが間違っているのかもしれない。しかし、完成できそうにない作品企画をしても意味はない。やはり作品を完成させることが優先だ。

新米B「私は、戦闘機による空中戦をやってみようと思いました。戦闘機なら関節もないし、動きのデザインがしやすいのではないのでしょうか」

先輩「おー。なんかものすごく優等生的な答えだ。でも空中戦というのは、背景が何もないので、動いていることを表現しにくいで」

新米B「ええ、そこで、戦闘区域をどんどん変えていくのです。最初は海、それから林になって、山岳地帯へ。ラストのクライマックスはグランドキャニオンです」

先輩「グランドキャニオンか。迫力のある映像ができ

るな。でも、短い作品でそんなに背景が変わっていくのは、ちょっと無理があるんじゃないか」

新米B「べつに最初からグランドキャニオンでもいいです。それから、最初から最後まで、ある戦闘機の視点の映像でまとめようと思うのですが」

先輩「それはだめだ。そういう奇をてらった演出は失敗の元。そんな制限をつけてしまうと、できる映像が限られたものになって、見ているほうもすぐ飽きてしまう」

新米B「そうでしょうか。斬新で面白いと思いますが」

先輩「斬新じゃない。使いものにならないから、誰もやっていないだけなんだ。ウソだと思ふのなら、実際にそんな映像ばかり作ってみればいい。30秒間見るに耐えるものができれば誉めてやるよ。その場合でも、他の視点から見た映像をおり混ぜるほうが、ずっとよくなる」

これは実際に作品制作をしてきた経験がものをいう。

新米C「私の考えたテーマは“不法駐車”です」

先輩「こりやまたわけのわからんこといい出したなあ」

新米C「問題になっている不法駐車を取り上げ、不法駐車によって事故が発生する過程を映像化するのです」

先輩「それはシリアスなのか？」

新米C「もちろんシリアスです。ちょうど政府広報のようなイメージです」

先輩「うーん、政府広報のCG化というのは、アマチュアCGA始まって以来の試みだろうな」

新米C「車や人体の形は、非常に単純化すれば十分可能だと思いますが」

先輩「技術的な問題点はない。発想も非常にユニークだ。だけど、そんなの作って何が面白いんだ？ Mは3人の意見を聞いてどう思う」

M「やはり、グランドキャニオンのドッグファイトがいちばんともだと思います。曲にも合わせやすいし」

ということで、電子バイクの路線がドッグファイトに変更することで意見が一致した。同時に、“カッコいい”“スピード感”という2つのコンセプトが決まった。

あとは、スタッフ内で自由にアイデアを出し、監督(または取りまとめ役)が、皆の意見をまとめていく。

柚姫の明るい悩み相談室

はじめまして、柚姫です。“ゆず”と書いて“ゆう”と読みます。今度、私、2代目姫を襲名しました。で、気分一新、コーナー名も変わりましたが、“あき姫の迷える子羊のコーナー”同様にかわいがってください。このコーナーの内容はいままでどおり質問・お便り中心です。ま、気が向いたらちょっと違うことも……。

最初ですし、少々自己紹介など。今年大阪大学に入学した新入生です。DōGAに参加するきっかけは……気づいたらいたんですけれど。ま、CGAで映画なんか作れるといいなあ、なんて思ってた。実は、実は、コンピュータなんてこれまで触ったことないんです。なのに、なのに、いきなり原稿は書かされる。来年の春にはCGAシステムがきつと、いやたぶん、いやもしかしたら、使えているはず……うるうる。けど、ご質問にはせーいっぱい(迷?)解答していきますのでよろしくお願ひしま〜す。

相談者：あの～、実は点光源の使い方なんですけど。3月号のあき姫のコーナーにあったとおりにやってみたのですが、どうもうまくいきません。私はどうしたらよいのでしょうか。姫：そ、そうなんです。ごめんなさい。あき姫に代わってお詫びします。実は3月号の解説にミスがありました。目ざとい方はもうお気づきと思いますが、左右の括弧の数が一致していません。

(書式)

誤 {mov(X Y Z) light point(rgb(R G B) L)}

正 {mov(x y z) light point(rgb(R G B) L)}

当然、(例)も間違っているわけですね。さあ、3月号と赤ペンを取り出して添削しちゃいましょう。

(例)

誤 {mov(10 20 30) light point(rgb(0.5 0.5 0.5) 200)}

正 {mov(10 20 30) light point(rgb(0.5 0.5 0.5) 200)}

というわけです。本当にごめんなさい。

相談者：私は“名なしのこんべえ”です。CGAシステムを申し込むのに、自分の住所も名前も書くのを忘れ、Oh!Xの連載の各読者通達事項のコーナーで全国指名手配されました。スタッフの皆さんお手数をおかけして申し訳ありませんでした。こんな私ですが、どうしたらよいでしょうか。

柚姫：そ～いうことは、お早めにご連絡ください。ちゃんと送ります。でも、あの～、名なしのこんべえさん。今回のお便りにもお名前、住所が書かれていないんですけれど……。

面白いお便りから、CGAシステム・DōGAプロジェクト・その他についての質問やお叱り、ご要望といった真面目なお便りまで何でもOK。どんどんお便りください。

「具体的に飛行機は何にするの？」
「サンプルデータのF16を流用すればいい」
「飛行機だけじゃつまらないから、陸上部隊も出そう」
「ややこしくなるから反対。敵味方の区別がつかないぞ」
「味方の兵器はすべて赤い色で統一しては？」
「赤いF16なんて見たくないよう」
「敵の戦闘機の数ほどのくらい？ 5機ぐらいかな」
「20機ぐらいの編隊も、かっこよさそうだけど、動きのデザインが面倒だな」
「機数はなんで戦闘をしているかの問題じゃないの？」
「短編なんだから、設定なんてどうでもいいと思うけど」
「F16に限らず、いろんな機種を出したら？」

「それも、ややこしくなりそう」
「でも、複葉機対F16なんて面白そう」
「あっそのアイデアいい。複葉機がF16をかたっぽしから撃ち落としていくんだ」
「F16と複葉機じゃ複葉機に勝ち目なんかないじゃんか」
「そこがナンセンスコメディなんだよ」
「おいおい、いつからコメディになったんだよ。かっこいい、スピード感じじゃなかったのかい」
このように、仲間同士うだうだ話し合っているうちに、いろいろアイデアが出てくる。監督が新しいアイデアに関心を示せば、元の内容など気にせずどんどん話を膨ら

寺田の教育的指導

「お客さん、えらい久しぶりやねえ！ へえー、5ヵ月ぶりになるか。長いこと、どこ行っちゃったんや？ え、高野山にこもって修行しとった？ うーん、そのわりには、ちょっとも文章上手なとらんな……。というわけで(?)、1年たったこの連載ですが、心も機も一転せず、相変わらずのこのコーナーの始まりです。さっそく本文へ突入だっ。

今回ご紹介する作品は、本職の日本画家(!)でいらっしゃる東京都の森山知己さん制作の“走る「MASAYA」”と“メリーゴーランド(特にタイトルはついていませんでした)”です。これは現在制作中の作品の一部で、完成すればMASAYA君がCRT上でおもちゃと遊び回るといふ作品になるそうです(これは楽しみ！ 次回のCGAコンテストの入選有力候補!)

モデリングのセンス、色の配置などさすがプロといった感じで、独特の可愛く楽しい雰囲気のあるものに仕上がっています。CGAコンテスト入選作の「デファインナブル・ファンクション」の色の使い方にもショックを受けましたが、この作品も新鮮な気持ちよさを感じました。いままでのCG作品は、どちらかというとメカニカルなものが多かったのですが、これからはこういう作品も増えていくことでしょう。

技術的な面では、この作品はこれまでの連載で解説してきた基本的なテクニックがすべてうまく使いこなされています。皆さんも、CGAシステムをこの程度にまで使えるように「精進」してください。まずモデリング、これは作る人のセンスに左右されますが、比較的簡単な立体の組み合わせでうまく処理しているところは参考になるでしょう(人体はちょっと違います)。私は木馬の目が可愛くて気に入ってしまいました。

次に、色・アトリビュートですが、台の部分の赤色と、木馬の白の組み合わせがとても鮮やかできれいですね。メインの色があり、それが引き立つように周りの色が選ばれている気がします(このへんは、さすがプロの画家さんですね)。

物体の動きも、MASAYA君の手足の動き、回転しながら上下する木馬など、構造体を使ってうまく表現されています。どうやってこういう動きができるか皆さんも考えてみてください(FSCに慣れるには、いろいろとやってみるのがいいでしょう。高校程度の数学の勉強もしてください)。

また、これとは別に「少し凝った」形状とい

うことで、“人体モデル”も送っていただきました。しかし、これは「少し」どころではなくかなり凝った形状データで、人体らしい特徴がよく出ています。面数が多いのは、人体という題材の都合上仕方ないですが……。

*

ところで、森山さんからいくつか質問が届いていますので、この場でお答えしたいと思います。

Q: RENDの実行のとき、データの数が多いと、コマンドラインに収まらないのです。

A: COMMAND.Xの制限で、255文字以上のコマンドは実行できません。たくさん物体を使って作画するとき、RENDに与える引数が多くなってこの制限に引っ掛かってしまいます。こういうときは形状ファイルや、アトリビュートファイルをいくつかまとめてひとつのファイルにしてください(ひとつのファイルに2つ以上の形状データが含まれていても、ちゃんとRENDは受け付けてくれます)。

多少面倒ですが、エディタを使うのが確実です。たとえば、A.SUFとB.SUFというファイルをひとつにまとめたときは、まずエディタ(ED.X)を立ち上げ、A.SUFを読み込みます。そして、F2キーなどで、ファイルのいちばん最後の行より下の行へカーソルを持っていき、そこでESC-YコマンドでB.SUFファイルを読みに読み込んでください。ESC-Tコマンドでファイル名を変更してから(拡張子はSUFとしてください)、セーブすれば作業は完了です。

3つ以上のファイルのときも同様のことを繰り返してください。できあがったファイルは、そのままRENDにかけられます。

Q: FFのオプションスイッチの使い方がよくわからないのです。

A: 森山さんのように、データが多く複雑な動きをするものを作るとなると、「メモリが不足しています(制御用)」とか、「メモリが不足しています(データ用)」などのエラーがすぐ出るようになります。読んで字のごとくワークエリアが不足していますので、オプションスイッチで増やしてください。

-scで制御用バッファ、-sdでデータ領域の大きさを指定できます。普通は、-sc10000 -sd100ぐらいで問題なく動作すると思います。足りなければ、もっと数字を増やしてください。また、デフォルトでは読み込めるフレームソースの大きさは、32Kバイトまでなので、それ以上に大きいときは、-sfオプションで大きさを指定してく

ださい。

あ、それとPESから使うときは、「オプション」のウィンドウは使わずに「フレームフィルタ」ウィンドウに「-sc10000」などを書いてください。RENDのバグ、じゃなくて仕様で2文字のオプションは「オプション」ウィンドウからうまく使えないのです(ご了承ください)。

*

さらに、森山さんからは非常にていねいなアンケートの回答と、プログラムの改良点、今後のCGAのあり方などに関する意見、そのうえビーム券まで送って頂きました(あまりD6GAにエサを与えないでください。調子に乗ってしまいます)。そのお礼というわけでもないのですが、森山さんにはCGAコンテストの作品集ビデオをお送りしたいと思います。

ついでに、今回は私の作ったプログラムBOMB.Xを紹介します(この号が出るころには、ネットにもアップしているはず)。このプログラムは、物体が爆発して飛び散っていくところを表現したくて作ったもので、「ボイジャー」やCGAコンテストのオープニングアニメにも一部使われています。

アルゴリズムは簡単で、まず物体データをバラバラにして、物体の1面1面を独立した形状データとして切り離します(つまり100個の面からできた物体なら、100個の「部品」に分かれるということ)。そして1つひとつの部品が飛び散っていく動きを計算して、それをフレームソースの形で出力します。それをFFにかけて、RENDでレンダリングすれば、物体が飛び散るアニメのできあがりです。写真を見てもらえば、何をやっているかよくわかると思います。

CGAシステムはデータの規格がきっちり決まっているし、FFやRENDといったしっかりしたプログラムがすでにあるので、このような安易なアルゴリズムのプログラムでも、結構面白いことができてしまいます。皆さんもいろいろと挑戦してみてください。私もあと2つ3つ「小ネタ」を考えているので、できあがったらまた発表したいと思います。

最近では送られてくる作品のレベルが上がってきて喜ばしい限りですが、少し数が減ってきています。作っている途中で行き詰まってしまったものや、こういう事をするにはどうしたらいいの？ といった質問もお待ちしています(そのほうが解説もしやすいですし、なんちゃって)。それではまた次回お会いしましょう。さようなら。

ませよう。要は監督が気に入るかどうかであり、スタッフがアイデアをたくさん出し、監督が取捨選択しながらイメージを固めていけばよいのだ。

監督「基本的なストーリーは、最初、複葉機がグランドキャニオンをのんびり飛んでいる。そこにF16の編隊が現れ、攻撃を仕掛けてくる。複葉機が反撃を仕掛けると、これがめっちゃくちゃ強い。F16を片づけた複葉機が、またのんびりと飛び去っていく（起承転結の構成がちゃんとできている点に注意）」

「見せ場は、ドッグファイトってことになるんだろうけど、複葉機はどうやって戦うんだろう？」

「プロペラの間からマシンガンを打つんじゃないのか」

監督「ちょっとまとも過ぎ。もうちょっと過激でいい」「ミサイル？」

監督「ミサイルはF16。複葉機は別の方法で戦わないと」「ちなみにF16のミサイルは熱線追尾型だから、プロペラ機には通用しません。エンジンの熱量が足りないんだ」

監督「事実なんて映像作品なんだからどうでもいい。F16はミサイルをたくさん発射しないと絵にならない」「複葉機から出ているタイヤで蹴飛ばすとか」

監督「面白いけど、いまいち」

「横になって、2枚の翼でF16の翼を切り落としたり」

監督「むちゃくちゃだけど、そんなのいいな。F16の方は5機ぐらいあって、初めの1、2機はけっこうまともな方法でやっつけるんだけど、だんだんむちゃくちゃになってきて、最後の1機は見ている人が“ウソつけ！”と怒鳴りたくなるほどエスカレートしていくんだ」

「複葉機が変形して、戦闘ロボットになるとか」

「なんだよ、アニパロかよ。オレも好きだけど」

監督「その路線はやらない」

「変形して、袴、ちょんまげ姿になったらうけるで」

監督「やらないものはやらない」

「ロボットとはいかなくても、変形して最新型の戦闘機になってもいいと思うけど」

監督「時代遅れの複葉機がF16を倒すことに意味があるので、最新メカになってはいけないんだ」

さて、作品企画も最終段階になってくると、監督がだんだんガンコになってくる。それは、監督の頭の中で自分のイメージがしっかりできた証拠であり、自信が出てきたことを意味する。“この監督は私のアイデアを取り入れてくれない”などとスネずに監督のつかみ始めたイメージを積極的に理解し、まとめる方向に持っていく。

監督「オープニングは、青空のグランドキャニオン。遠くのほうから複葉機が飛んでくる。ふらふらと、いかにもオンボロって感じ。ロングの長回しで（長回しは1カットが長いこと）。そしてタイトル。そのあとBGMの雰囲気は激しくなってくる。空の片隅に何かが現れる。なんだ？」と思わせて、いきなりF16のアップ。ポンッポンッと短いインサートカットを入れて、緊迫感をあおる……」

最後に監督から作品の解説をしてもらう。どんなストーリーということだけではなく、各シーンがどのようなイメージなのかということが重要となる。全スタッフが、監督のいわんとするイメージを共有しなければいけない

のだ。

以上が作品企画会議のようすです。このあと、監督は担当者といっしょに絵コンテの作成に入り、他のスタッフは、資料集め、形状データのデザインなどを始めます。監督は、自分のイメージをしっかりと固めて、全カットの細部まで決定していかなければいけないのですが、アイデアに詰まったりすれば、何度でも企画会議を開くことが必要です。なお、この作品の制作過程については（ちゃんと制作が進めば）また次回にでも報告しますのでお楽しみに。

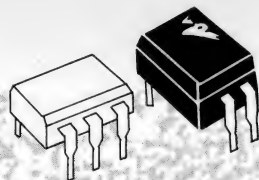
／ おわりに

最後にもうひとつ企画に大切なコツの話で締めくくります。それは、“365日、24時間企画をする”ことです。いきなり“さあ作品企画をしよう”という思いつくなんてものではないので、常日頃考えておくのが大切なのです。とはいっても、いろいろ忙しい毎日、作品企画ばかり考えているわけにもいきません。ですから、思考の5%ぐらいだけを常に企画に回しておいて、何を見、何を聞き、何をさわっても、一度“これは作品になるんじゃないか”と想像してみるのです。朝、食卓について箸を手にとったら“箸の形って、CADで作りやすいな”と気がつきます。そして“食卓ってのはいろんなキャラクターが集まった集会場だな”と見回すのです。“つま楊子でフェンシングなんて楽しそうだな”と、赤と青の箸がつま楊子でフェンシングをしている映像が目には浮かべば完璧です。

何もその場でストーリーを作れというのではなく、アイデアの蓄積が大切なのです。試しに明日の朝、家を出て学校（あるいは会社）に着くまでの間、“これは作品になるんじゃないか”を試してください。ネタの1つや2つ必ず見つかるでしょう。

作品企画は、作品制作の中でも楽しく、盛り上がりるところです。あまり難しく考えずに、積極的に楽しむのも大切でしょう。今回解説した企画法はあくまで、これをヒントにしてくださいといった程度のもので、これを読んで、皆さんも作品を制作する気になっていただければ幸いです。

さて、「FAR SIDE MOON」のウイルス事件ですが、当スタッフも購入した者がいたため、一時期すべてのディスクの持ち出し、発送を禁じるの騒ぎが起きてしまいました。しかし、ウイルスを確実に見つける方法も確立していない現状では、当チームも同様のトラブルを起こさないという保証はありません。憎むべきはウイルスを作った犯人であり、ワクチンソフトの無償配布など多大な損害を出したアートディンクは最大の被害者だと思えます。悪意を持ったウイルスによって、「FAR SIDE MOON」の売り上げが落ちたり、アートディンクが経営の危機に陥ったとしたら、私たちユーザーにとっても不幸であり、ある意味でウイルスに屈したことになるような気がします。ですから、アートディンクからワクチンを受け取ったユーザーは、実費だけでもアートディンクにカンパしてはいかがでしょうか。



基本インタフェース回路 その1

Misawa Kazuhiko
三沢 和彦

いよいよ今月からハードウェア工作入門の始まりです。理論編、工作編、活用編の3回で1つのテーマを扱っていきます。最初は部品の少ない簡単な工作でI/Oの基礎から解説していきますので、ぜひとも続けて読んでいってください。

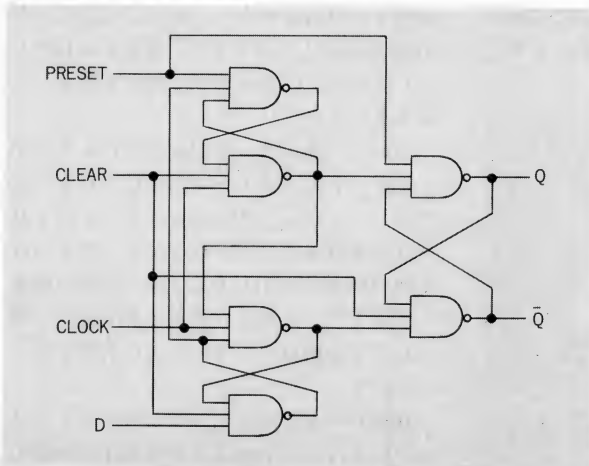
先月号の予告編でお伝えしたように、さっそく自作ハードをX68000のジョイスティックポートにつなぐことを考えてみましょう。まず初めに、どうしてジョイスティックポートにつなぐと外部機器とデータのやりとりができるのか皆さんは理解していますか？ それは、ジョイスティックポートがI/O(Input/Output)インタフェースの役割を果たしているからです。今回は、連載第1回ということで、まず、インタフェースの概念とその仕組みというもっとも基本的な事柄から説明していきたいと思います。

インタフェースの基礎とジョイスティックポート

そもそも「インタフェース」という言葉は「境界面」という意味で、コンピュータの世界では、異なる2つのハードウェアを仲介してつなぐ装置のことをいいます。特にコンピュータ本体と外部機器とを接続するのに使われる場合をいうことが多いようですが、基本的には、キーボードやディスプレイといった入出力機器とCPUとをつなぐための内部回路はすべてインタフェースというべきです。

このインタフェースの重要な役割は、

図1 D-フリップフロップ

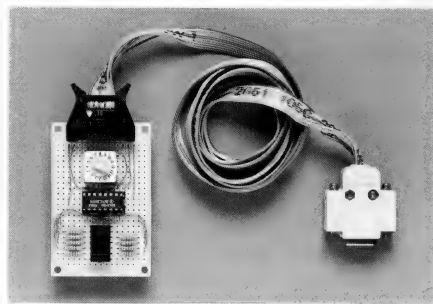


- 1) データを受け渡しする機器どうしの間で互いにやりとりができるようにデータ形式を変換してやる。
 - 2) 時間で変動していくデータを処理できるように一時的に保持してやる。
- の2つです。これらの点について、少し具体的に突っ込んでみましょう。その前にコンピュータ内部でのデータ形式について復習しておきます。

データは基本的に1（真）か0（偽）で表し、これを1ビットといいます。ただし、X68000は16ビットまとめて処理します。ただし、インタフェース回路は通常8ビット単位で処理します。8ビット（16ビット）まとめという場合には、信号線が8本（16本）横に並んでいて、それぞれの信号線上に各ビットのデータが乗っていると考えればよいのです。このような横並びのデータ形式を「パラレル」データといいます。

コンピュータ内部ではこの横並びの信号線をバスラインといい、CPUからメモリ、そしてキーボード、CRTディスプレイ、ディスクドライブなどの各回路につながっていて、隅々までデータを流します。このとき、たとえ1/0の2進データをそのままCRTに送ったとしても、画面上に文字は表示できません。画面上に文字を表示するには、ASCIIコードのような2進データをCRT用のRGB信号に変換してやってからCRTに送らなければなりません。

また、キーボードからの入力にしても、我々が入力するデータは各々の文字に対応するキーを押すだけで、キーの押された場所に対応してそれぞれ2進の文字コードデータに変換してやらなければCPUは処理できません。これらがすなわち、1)というデータ変換の意味



製作するインタフェース回路

です。

次に、キーボードから文字が入力され、文字コードに変換されてバスラインにデータが乗ったとしましょう。このとき、CPUは何をしているのでしょうか。

CPUはたえずデータをやりとりしたり、あるいはCPU内部で演算している最中だったりするかもしれません。すると、CPUにしてみれば、いつキーボードからの入力があるかわからない状態で、突然文字データが送られてきても、読み過ぎてしまうのです。

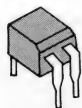
そこで、2)でいうように、送られてきたデータを一時保持しておいて、CPUが都合のよいときにおもむろに読み出してくるようしておく必要があるわけです。また、外部機器にデータを出力するときも、外部機器の読み取りタイミングに合わせるためにデータの保持が必要となることが多くあります。

さて、どんな入出力データでも、CPUが処理するのがパラレルデータである限り、インタフェースは基本的にパラレルデータを扱わなければなりません。このようにパラレルデータと他のデータ形式とを仲介するインタフェースをパラレルインタフェースといい、このパラレルインタフェースこそがインタフェースのもっとも基本的な形だといえます。

パラレルインタフェースの実際の回路は、図1に示すD-フリップフロップをバスラインのビット数分だけ横並びにしたものが基本です。この回路は、データの保持だけの

機能しかなく、データの変換は行えません。しかし、データ形式を変換するインタフェイスでも最後は必ずCPUのためにデータを保持しておかなければならないので、すべてのインタフェイスがこの部分を持っています。それだけ基本的なインタフェイスですから、専用のICにもなっています。

もちろんX68000の中でもパラレルインタフェイスはあらゆるところで活躍していますが、その代表的なICが8255AというICなのです。そして、問題のジョイスティックポートも、この8255Aそのもののなのです。



ジョイスティックポート 8255Aの機能

それではここで、8255Aの機能について説明しましょう。8255Aは最大8ビットずつ3組、計24ビットの入出力をプログラムによって設定できます。もう少し詳しくいうと、このICはCPUのバスラインにつなぐ8ビットの端子と外部機器につなぐ8ビットの入出力信号端子（ポート）を3組持っていて、このポートをそれぞれ入力として使うか出力として使うかを外部からの命令によって指定でき、このIC1個使うだけで3組のインタフェイスとして機能させられるわけです。

実際の信号の入出力は簡単で、入力にはポートの端子に5VをつなげばH、GNDレベル（0V）をつなげばLのデータとして、CPUが取り込みます。出力は逆に、CPUがHのデータを送ると端子に5Vが、Lには0Vが出てきます。そして、このインタフェイスとしての機能には、3種類のモードがあるのですが、初心者の方は、これから説明するモード0のみ理解しておけばよいでしょう。ジョイスティックポートもこのモード0で使っています。

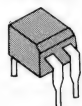
モード0では、ポートA、BおよびCの半分ずつを任意にそれぞれ入力または出力ポートとして設定できます。そして、入力を指定したときはその入力時にA、B、C

の各ポートに加わっているデータがCPUに読み込まれます。ただしこのとき、ポートのデータは保持されないで注意が必要です。それに対し、出力を設定したときには、CPUから各ポートに出力されたデータは保持され、次にデータが出力されるまでは同じデータが出力され続けます。

表1にジョイスティックポートの各端子の信号の内容を示します。これは、X68000の取扱説明書の表をそのまま転載したものです。備考欄を見てわかるとおり、ジョイスティックポートの信号は+5VとGND以外は、単に8255Aのポートが直結されているだけです。ジョイスティックポートの端子No. 1~4はポートAを入力に指定して、4ビットの入力専用、No. 6~8はポートCを出力に指定しての3ビット出力専用のインタフェイスになっています。

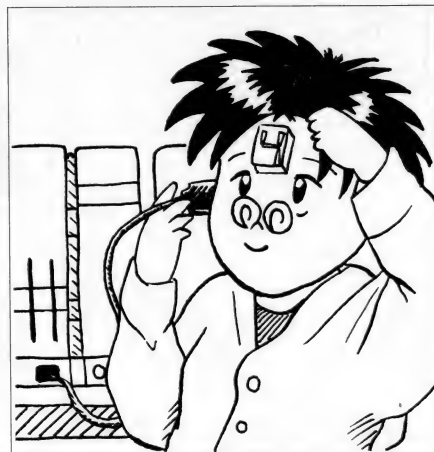
なお、端子No. 6と7とは入力出力両用になっていますが、実際はCPUからポートCにデータを出力してしまえば、そのデータが保持されて、端子No. 6と7とに出力され続けます。ただ、端子No. 6と7の出力は、端子4の出力と反転しているの、注意が必要です。このあたりの事情は、プログラムを組むときに問題になるので、再来月にX68000を実際に使って説明することにします。

以上の説明でインタフェイスの基本的な考え方となぜジョイスティックポートが、インタフェイスとして使えるかが理解できたと思います。それでは、次に今回の回路を挙げ、ハードウェアとして実際にパラレルインタフェイスをどう使っていくかを説明していきたいと思います。



LED & SW 基本インタフェイス回路

今回のテーマは基本的な入出力ということで、外付けのスイッチからの入力を取り込む回路とポートからの出力をLEDで表示させる回路とを作ってみようと思います。



入力4ビット、出力3ビットなので、入力については0~15の数字をロータリースイッチという回転式のスイッチからひとつ選んで、それを2進データに変換して取り込む回路とし、出力については0~7の10進数をひと昔前の電卓などに使われていた7セグメントのLEDというものに表示させる回路とします。

実際の回路は図2のとおりですが、なんとICは1個しかないので、最初の回路としては、部品点数が少ないほうが回路を理解しやすいし、工作も楽だと思います。それでは、入力部と出力部とに分けてそれぞれ詳しく説明していきます。

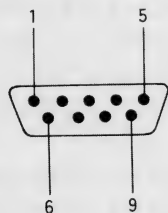
●入力部

まず入力部ですが、これは16進ロータリースイッチ1個だけという、なんと工夫のしようがない回路です。それだけにスイッチの選択を誤ると、予想どおりの動作はしてくれないので注意が必要です。この16進ロータリースイッチには、5つの端子がついていて、そのうち1つが共通端子で、あとは2進4ビットの端子になっています。スイッチを0~15の16通りを選べると、選んだ数を4ビットの2進数にスイッチのなかで変換し、ビットの1になる位の端子を共通端子と導通させます。通常は共通端子をGNDに落としておきます。言葉ではわかりづらいので、図3に端子の動きを例示しておきましょう。

これらの端子は、そのままジョイスティックポートのNo. 1~4に直結されているだけです。ジョイスティックポートのそれぞれの入力端子に何もつながらないときには、X68000の内部でHになるように回路が組まれています。そして、各端子をGNDに落としたとき初めて、Lの入力に対応しているのです。

16進ロータリースイッチから出てくる2進データはビットが1になる端子だけがG

表1 ジョイスティック用コネクタ(アタリ社規格準拠)



端子No.	信号名	I/O	備考
1	IOA0	In	8255のPA0端子
2	IOA1	//	// PA1 //
3	IOA2	//	// PA2 //
4	IOA3	//	// PA3 //
5	Vcc1	Out	+5V
6	IOA5	I/O	8255のPA5/PC6端子
7	IOA6	I/O	// PA6/PC7 //
8	IOC4	Out	// PC4端子
9	GND	—	

ND につながり、ビットが0になる端子はどこにもつながらないようになっているので、ロータリースイッチから出てくる2進データがそのままジョイスティックポートに取り込まれることになるのです。

このとき注意しなくてはならないのは、各ビットの1のデータが入力のL、0のデータがHに対応している点です。このように正負が反転しているデータを正しく読み取るのはソフトウェアのほうで対処しますが、詳しいことは再来月のプログラム実習編で解説します。

●出力部

出力部には、IC (LS247) 1個と表示用の7セグメントLED (TLR313) 1個とが使われています。7セグメントLEDは、図4のように、7本の線状LEDが「日」の字型に並べられているものです。この7本の組み合わせで10進数の0から9までを表現します。

ジョイスティックポートからは2進3ビットのデータが出力されてきますが、これをそのまま7セグメントLEDに入れてやってもだめで、7セグメントLEDに表示させるためにはデータを7本のセグメントの組み合わせに変換し直さなければなりません。LS247というICは、2進4ビットのデータを10進数に変換し、さらに、その10進数を7セグメントLEDに表示させる専用のICです。最初に述べたインタフェースの役割のうち、出力のためのデータ変換を一手に引受けているのがこのLS247ということになります。そして、ポートの出力をそのままLS247の入力につなぎ、LS247の出力をそのままTLR313の各ピンにつなぐだけで簡単に表示部が実現されます。

ところで、LS247とTLR313との間に抵抗が入るのですが（図路図では省略）、これ

は、LEDに大きな電流が流れすぎて壊さないように保護するものです。

*

以上で、回路図の説明は終わりです。意外と簡単だと思いませんか？ まだわかりづらいという皆さんにも、来月の製作実習編、再来月のプログラム実習編で、まったく同じ回路について実際に動作を確認しながらくどいほど繰り返し説明を加えていきますので、ご安心を。次回の製作編では、部品表と実体配線図を載せて、初心者でも完成できるように丁寧に解説します。それまでに工具を揃えて待っていてください。では、来月。

図2 回路図

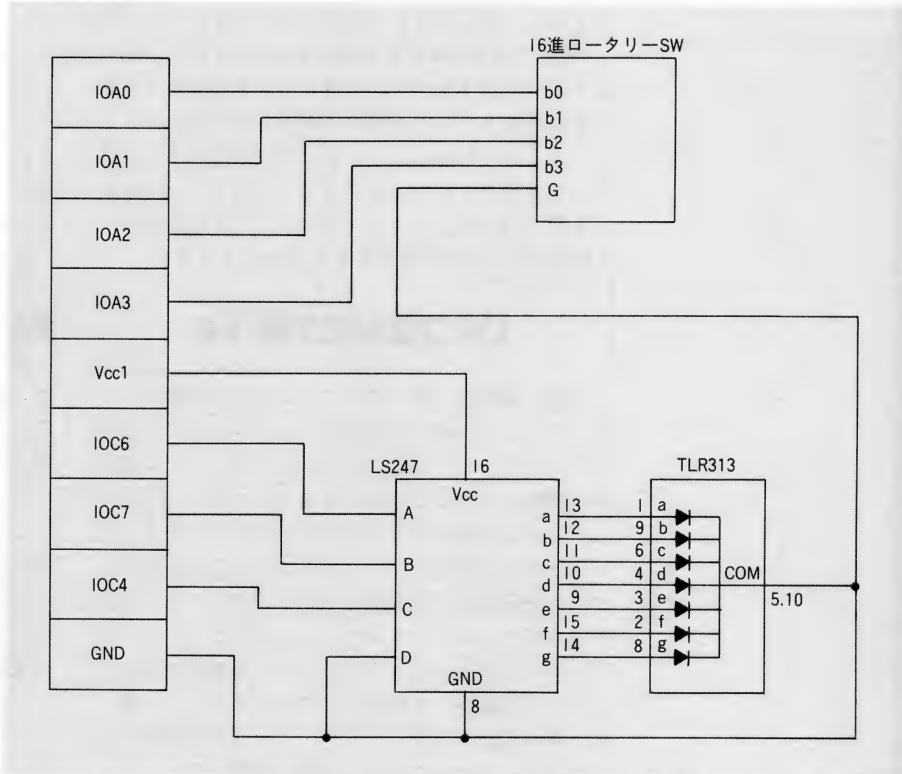


図4 LEDの表示

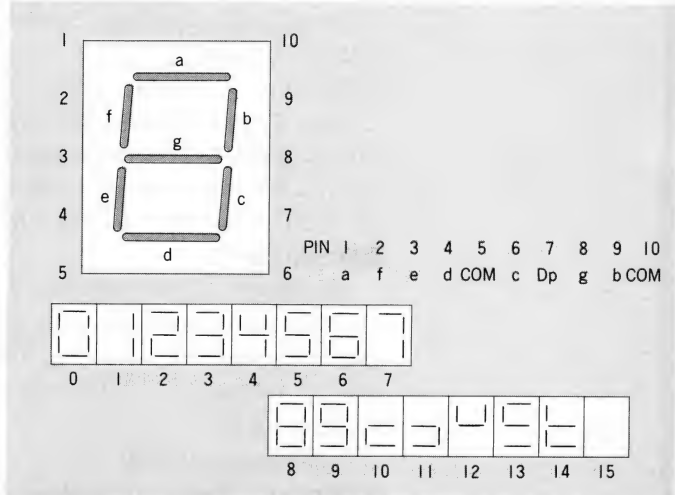
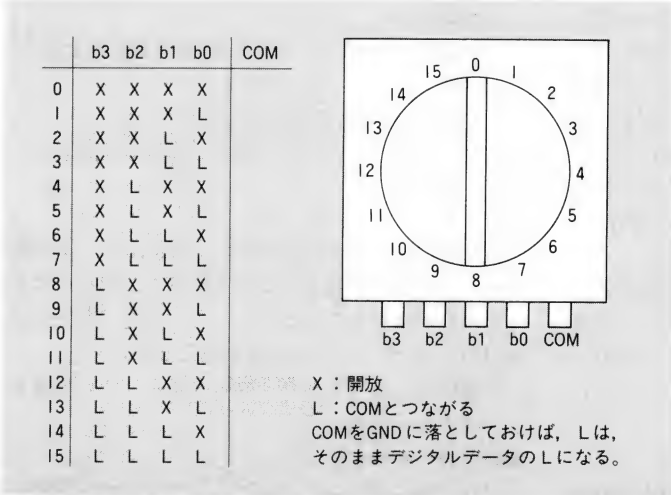


図3 スイッチ



パズルゲームを作る(後)

Izumi Daisuke 泉 大介

さあ、Yet Another Columnの後編です。揃ったブロックを判断して消すアルゴリズムを考えてみましょう。また、ディスク版はコンパイラ版でしたが、インタプリタのままで十分なゲーム性を持たせるための工夫もポイントです。

先月号の付録ディスクでお届けしたYETはいかがですか、楽しんでいただけていますか。編集室ではさらにスコアが伸び、現在の最高得点は(よ)嬢の42315点です。もうこうなると作者の私も手も足も出ません。げに恐ろしきはゲーマーかな。

今月はYETの後半をお届けします。先月は、BGを使ったほうがキャラクタを使うより多彩なキャラクタを表現でき、しかも素早く表示することができるということを中心に、ピコピコゲームの基本のようなことをやりました。今月は「3つ以上のタイルが並んだら消える」というアルゴリズムを中心に、BASIC版のYETが出来るまでをお送りします。

いくつ並んだか調べる

先月、皆さんも考えてみてくださいとお願いしておいた「いくつ並んだか調べる」アルゴリズムですが、いいアイデアが浮かびましたか? ゲーム盤がある関数に与えると消去可能なタイルがすぐにわかるというようなものと思ったのですが、そうまい方法はありません。かといって任意のある時点のゲーム盤を端から順に調べていくのは、いかにも面倒そうです。コンパイルするならともかく、インタプリタではゲームにならないくらい時間がかかってしまうでしょう。それぞれのタイルについて、縦、横、斜めの計4方向につながっている枚数を調べなければならぬのですから大変な作業です。

そこでタイルが上から落ちてくる時の様子を考えてみましょう。このとき、下に積もっているタイルは3つ以上の同じ色が並んでいることはないはず。ということは、新しいタイルが落ちてきたことによって3つ以上同じ色が並んだところがあったかどうかを調べるだけで事足りるのではないのでしょうか。これで、ゲーム盤を端から調べるという作業が、3つのタイルを調べるだけでOKになります。

●縦方向に調べる

以後3連のタイルの一番上のタイルの座標を基にして考えていくことにしましょう。落ちたときの座標をx, yとすると、縦方向のチェックはここから下向きに、y座標を大きくしながら調べていくことになります。まず、

```
tile=bg-get(1,x,y)
```

で変数tileに、座標x, yに表示されているBGを読み

出します。ここで読み出されるのは、256×パレットブロック+1というコードです。これはbg-putでパターンを指定するときのコードと同じです。同じ色のタイルなら、コードは同じはずですから、

```
i=0
while tile=bg-get(1,x,y+i)
  i=i+1
endwhile
```

というループを実行すれば、同じ色のタイルが何個並んでいるかが変数iに求まります。iが3以上なら、並んでいるタイルを消せばいいのです。

もし3つ以上並んでいないなら、今度はtileにx, y+1のタイルをセットして下向きに調べます。

●横方向に調べる

横方向は左と右の両方を調べなければならないため、縦方向ほど簡単にはできません。とはいっても、まず左側につながっている個数を調べ、次に右側につながっている個数を調べて加えるというのはいかにも間抜けな気がします。

ここで採用した方法はまずどちらか一方へたどっていき、タイルが違う色になったらそこから逆方向に同じ色のタイルの数を数えていこうという方法です。具体的にはまず一時的なx座標を保持するための変数tmpXを用意し、

```
tile=bg-get(1,x,y)
tmpX=x
while bg-get(1,tmpX,y)=tile
  tmpX=tmpX-1 /* まず左側に移動する
endwhile
tmpX=tmpX+1 /* 行きすぎた分を戻す
i=0
while bg-get(1,tmpX+i,y)=tile
  i=i+1 /* そこから右向きに数える
endwhile
```

とします。

最初のwhile~endwhileのループではタイルを数え始める座標を移動させているだけで、何枚つながっているかは考えていません。2つめのwhile~endwhileでつながっている枚数を数えます。

この作業を落ちてきた3枚のタイルについて繰り返せばチェック終了です。

●斜め右、斜め左

これも横に調べるときと同じようにすれば簡単に

チェックできます。左斜め上の方向を調べるなら、

```
tile=bg-get(1,x,y)
tmpX=x : tmpY=y      /* xとyの2つを
                        用意
while bg-get(1,tmpX,tmpY)=tile
  tmpX=tmpX-1      /* 左斜め上に
  tmpY=tmpY-1      /* 動かしていく
endwhile
tmpX=tmpX+1      /* 行きすぎた分を
tmpY=tmpY+1      /* 修正する
i=0
while bg-get(1,tmpX+i,tmpY+i)=tile
  i=i+1          /* 右斜め下に動かしていく
endwhile
```

でOKですね。左斜め上から右斜め下へ同じ色のタイルが何枚つながっているかをこれで知ることができます。3連のタイルそれぞれについて繰り返すこととなります。あとは右斜め上の場合をリストを見る前に考えてみてください。

枚数チェックの汎用化

ここで紹介した方法は上から落ちてくる3連のタイルしか考慮していません。タイルが消去された場合、その上に積もっていたタイルは自由落下し、その結果新たに3つ以上のタイルが並ぶとこれもまた消去されます。これら2つの状況は非常によく似ています。うまく同じ方法で処理できるようにならないでしょうか。

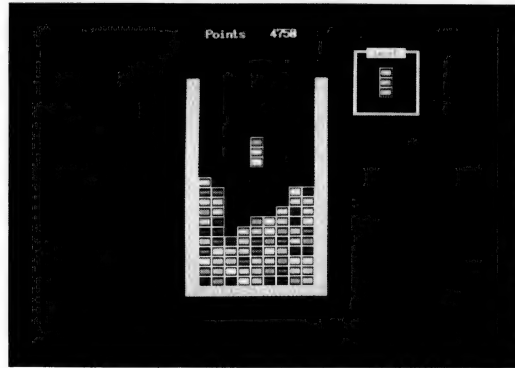
3連のタイルを、3枚のバラバラのタイルが「たまたま」塊になって落ちてきたのだと考えてみましょう。たまたま塊になって落ちてきただけですから、一番上のタイルの座標を基準にはほかのタイルの色のつながりを調べようという上の方法は訂正しなければなりません。タイルの座標を1枚ごとに配列に保存し、枚数チェックはこれを基に行うことにしましょう。

具体的には次のような要領です。配列名をdoubtとしておきましょう。3連のタイルが下に落ちたら、

```
doubt(0,0)=x : doubt(0,1)=y
doubt(1,0)=x : doubt(1,1)=y+1
doubt(2,0)=x : doubt(2,1)=y+2
```

と、3つのタイルの座標を配列に収めます。このとき、doubtEnd=2として最後のタイルの座標が入っている場所を表しておきます。同色のタイルが3つ以上並んでいるかどうかのチェックは、

```
for t=0 to doubtEnd
  tmpX=doubt(t,0)
  tmpY=doubt(t,1)
  tile=bg-get(1,tmpX,tmpY)
  /* 縦のチェック
  i=0
  while tile=bg-get(1,tmpX,tmpY+i)
    i=i+1
```



```
endwhile
  ~ 横のチェック ~
  .....
  ~ 左斜めのチェック ~
  .....
  ~ 右斜めのチェック ~
  .....
next
```

で行うことができます。for～nextのループでdoubt配列からタイルを1つひとつ取り出し、それぞれの方向についてチェックするわけです。

ここまではOKでしょうか。では次に消去する方法について考えておくことにします。ディスク版YETではタイルはシュワッと色が薄くなって消えていきますね。このような消し方を実現するために、3つ以上つながっているタイルの座標を保存しておき、消すときにはこれらの座標に表示されているBGに対して一斉に処理を行うという方法を採用しました。

消せるタイルの座標は、doubt配列の未使用部分に保存しています。0～doubtEndまでは、消せるかもしれないタイルが入っていますね（これがdoubtの由来だったりする）。その後ろに付け加えるかたちで、消せるタイルの座標を蓄えていきます。縦のチェックで具体例をお見せしましょう。まず、

```
clearEnd=doubtEnd+1
```

として、消去可能なタイルの最後をセットしておきます。

そして縦のチェックの部分を、

```
i=0
while tile=bg-get(1,tmpX,tmpY+i)
  doubt(clearEnd+i,0)=tmpX
  doubt(clearEnd+i,1)=tmpY+i
  i=i+1
endwhile
if i>2 then clearEnd=clearEnd+i
```

とするわけです。タイルが何個つながろうがおかまいたしに、とにかくdoubt配列にタイルを登録していきます。そして、最後にタイルの枚数(つまりi)を調べ、3以上ならclearEndを更新するわけです。もしタイルが2枚しかつながっていなかったのならclearEndは更新されませんから、次に行われる横のチェックでは再びdoubt(clearEnd,0)から使われる

というわけです。

●checkとeraseの意味深な関係

さてこれがタイルの色を調べるアルゴリズムのすべてです。YETのプログラム中ではこの部分はcheckという名前の関数になっています。そして実際にタイルを消去するのがerase関数です。

これら2つの関数は密接に関係しあっています。まずcheck関数が色を調べ、消去できるタイルを登録します。erase関数はそれを画面から消去しますが、そのとき消去するタイルの上にあったタイルを下へ移動させます。同時にこの崩れ落ちたタイルの座標を再びdoubt配列にセットするのです。doubtEndもこのとき更新されます。そしてerase関数は、崩れ落ちたタイルがあった場合は1を、なかった場合は0を返します。したがって、

```
repeat
  check( )
  chkFlag=erase( )
until chkFlag=0
```

というループにすれば、落ちるタイルがなくなるまでチェックと消去を繰り返すという処理ができます。

チェックの高速化とビット演算

さて、インタプリタではタイルが落下したあとのチェック時間が結構かかってしまい、このままだとゲーム意欲をそがれてしまうほどです。そこでチェックルーチンの高速化を考えてみました。

上から青のタイルが3連で落ちてきた場合を考えてみましょう。check関数は一番上のタイルから調べ始めます。まずは縦方向のチェックです。ここで同じ色のタイルが3つ並んでいることがわかり、消去するためにdoubt配列に登録します。これは先ほどから説明しているとおりですね。続いて横と斜めのチェックをし、一番上のタイルに関するチェックは終了です。

次に2番目のタイルについて調べ始めます。まずは縦方向からです。ところが縦方向は一番上のタイ

ルを調べたときにチェック済みのはずで。青の3連タイルなので、下向きに調べるのは無駄というものです。こういった無駄を省いてやろうというのがここでの高速化です。タイルが崩れたあとはさまざまな場所でタイルが落下しますので、すべての方向についてチェック済みかどうかを調べるのはかなりの効果が期待できます。また実際、この処理によってかなり反応がよくなりました。

では実際にどうやってチェックを行うのかを見ていきましょう。ここでは新しい考え方であるビット演算を使います。ビット演算は2進数で考えるとよくわかります。

```
for i=0 to 10
  print bin$(i)
next
```

を実行してみてください。bin\$関数は引数を2進数で表示する関数です。

2進数は0と1の2つの数字だけで成り立っています。0, 1の次は、1の位をそれ以上大きくできないので繰り上がり10になるのです。このままでは位取りが見づらいので、2行目を、

```
print right$("000"+bin$(i),4)
```

に変えてみてください。今度は頭に0が補われ、桁のはっきりわかるようになったと思います。

ではビット演算を紹介しましょう。次のプログラムを入力してみてください。

```
10 end
20 /*
30 func prbin( x )
40   print right$("000"+bin$(x),4)
50 endfunc
```

これは数字を4桁の2進数で表示する関数です。関数を作りたいだけなので、10行でいきなりendになっています。これを使って実験してみましょう。

X-BASICで使えるビット演算にはnot, and, or, xorの4つがあります¹⁾。

●not

notは単項演算子として使います。これは2進数の第n桁が0なら1に、1なら0にする働きがあります。

```
prbin( not &B0011 )
```

を実行してみてください。

```
1100
```

と表示されますね。

●and

これは二項演算子として使います。実例を見てみましょう。

```
prbin( &B0101 and &B0011 )
```

と実行してみてください。

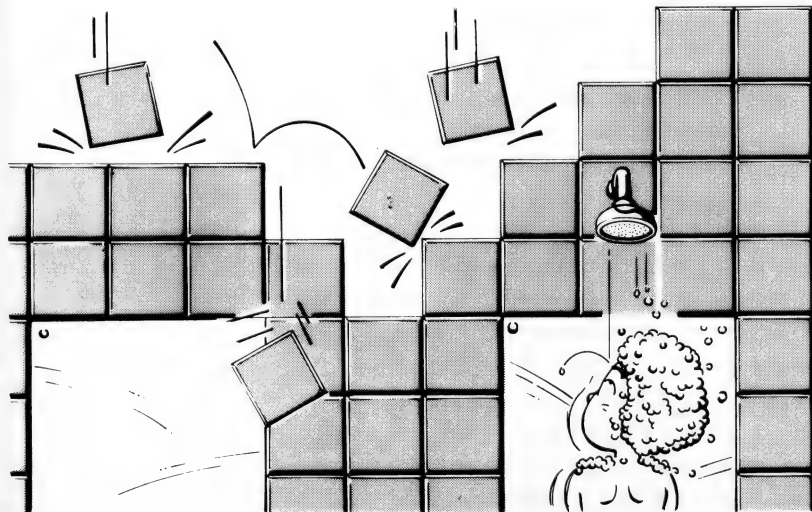
```
0001
```

と表示されましたね。わかりやすいように並べて見てみましょう。

```
0101
```

```
0011
```

1) ビット演算にはほかにもいろいろあるのですが、実際にはnotとandさえあればどんなビット演算でも作ることができ、X-BASICはさらにorとxorまで持っているのですから困ることはないでしょう。



0001

なにか気づきませんか。そう、第n桁がどちらも1の桁だけが1になっていますね。andは目的の桁が1かどうかを調べるのに使えます。1~10の中で2桁目が1の数を知りたければ、

```
for i=1 to 10
```

```
prbin( &B0010 and i )
```

```
next
```

とします。2桁目が1以外なら0000と表示されます。

●or

これも二項演算子として使います。実例を見てみましょう。

```
prbin( &B0101 or &B0011 )
```

を実行してみてください。

0111

と表示されます。orはどちらかの桁が1ならばその桁は1になります。そこで目的の桁を1にしたいときに使えばいいということになります。逆にandは目的の桁を0にしたいときにも使うことができます。

●xor

今回は使っていないのですが、ついでに説明しておきましょう。

```
prbin( &B0101 xor &B0011 )
```

を実行してみてください。

0110

と表示されますね。xorはどちらか一方の桁だけが1のときだけその桁が1になるのです。何に使うのか疑問に思われるかもしれませんがね。いま表示された答え0110を使って、

```
prbin( &B0101 xor &B0110 )
```

としてみてください。どうですか？ ついでに、

```
prbin( &B0011 xor &B0110 )
```

も試してみましょう。

●高速化の実際

check関数ではandとorを使って、いま見ているタイルがすでにチェックされているかどうかを調べています。リストを見てください。2680行でBGのテキストページ0を0で埋めつくしています。タイルを表示しているのはテキストページ1ですから、未使用のページ0をすでにチェックされているかどうかを調べるのに使うことにしました。座標x, yに表示されているタイルのチェックが終わったら、テキストページ0の座標x, yに印を付けるわけです。

この印は各方向のチェックの最初で調べます。2780行を見てください。tmpX, tmpYというのは、これから調べようとしているタイルの座標です。テキストページ0の該当座標からデータを取り出し、1とandをとっていますね。1というのは&B0001で1桁目が1です。これは縦のチェックが終わったという印に使っています。もし1桁目が0なら、andの結果は0になりますから条件が成立し、縦のチェックが始まるというわけです。

続いて2830行を見てください。ここはwhileで同じ色が続く限り下向きにタイルを調べ続ける部分です。

この2830行で調べ終わったタイルに印を付けています。orを使ってべつの桁を壊さないように印を付けている点に注意してください。

続く横のチェックでは印として2(&B0010)を、斜めのチェックでは4(&B0100)と8(&B1000)を使っています。

タイルの消去と高速化

高速化という点については、タイルのチェックよりこの消去のほうが実行価値があります。なんせこちらは消したあとのタイルの崩れという面倒な処理が待っていますからね。

check関数の実行が終わると、画面から消去すべきタイルはdoubtEnd~clearEndの間に入っていることになります。まずはこれらのタイルを画面から消さなければなりません。ディスク版YETではシュワッと消してみました。インタプリタでは速度の問題があつてあれほど気持ちよくは消えてくれないのです。そこで白い星を表示し、「消えた!」というイメージを出してみました。リストでいうと3640行です。次の3650行が高速化を行っている部分です。

erase関数の頭、3550行を見てください。colsという配列が宣言してありますね。これは消えたタイルのx座標を保持しておくための配列です。この配列を調べれば、タイルが崩れたために残りのタイルを落下させなければならない場所がわかるというわけです。すべてのタイルを「これは落下するのかな?」などと調べていったのでは時間がかかってしかたありません(実は初期バージョンは馬鹿正直にそうやっていたのですが……)。ここでは消去したタイルのx座標に対応するcols配列に1を入れ、消去の印としています。

このあとに続く崩壊を処理しているのが3860行からのfor~nextループです。ここではタイルが消えた列(これはcols配列を調べればすぐにわかる)を下から上向きに調べ、星印と空白を飛ばしてタイルを下へ詰めていく処理を行っています。3950行のif文のorの前後が星印と空白のチェックをしている部分です。崩れ落ちたタイルはdoubt配列に登録され、再びcheckループへと戻っていくことになります。

速くなる、速くなる

ゲームが進んでいくと速度がだんだん速くなっていくというのはこのテのゲームの常です。これを仕切っているのが60行で宣言されているlevels配列と、現在のレベルを保持している変数levelです。levels配列にセットしてあるのはレベルに対応した点数で、点数が300点より大きくなればレベルは0から1へとアップします。ここはディスク版と異なっている点です²⁾。

速度を上げる方法については1420行のinitFM関数を見てください。先月のものに手を加えてあり、

2) ディスク版では出現したタイルの総数によってレベルが上がっていくようになっています。つまりディスクではタイルがあまり落ちないうちに得点をあげておくのが高得点への道となるわけです。

30本のトラックを確保するようになっていきます。このうち1~20のトラックに、@L192R、@L182R、……と次第に短くなっていく休符がセットしてあるのです。先月休符を演奏して待ち時間を稼ぐという方法を紹介しましたね。レベルに応じて鳴らす休符を次第に短くしてやれば、待ち時間は次第に短くなりタイルはどんどん速く落ちるようになるわけです。ポイントに応じてFM音源のチャンネル1に休符をセットしているのがerase関数の中の3770行にあるif文です。

リストの入力方法と改造

リストは先月のものを基本にしています。足りない部分を補い、check関数とerase関数を補った形になっています。補った部分は「/* *」で始まるコメントがついています。これらの行を追加し、2つの

関数を入力すればOKです。先月号のリストを入力してある方は、今月号のリストと見比べながら補っていきましょう。

今月はリストの詳しい説明は行いませんでした。これは実際のリストを追いつながらではアルゴリズムを十分説明できないと判断したからです。まずはアルゴリズムを頭に置いて、それがどのようなプログラムになっているのか追いかけてみてください。

levels配列は少々やさしく設定してあります。慣れてきたらスピードが上がっていくのが遅く感じることでしょう。最初の方は100点ずつレベルが上がっていく程度でいいかもしれません。YETで高得点を取るためにも、自分でlevels配列に手を入れてスピードを調整してください。

さて来月は「ファイル処理」をやりたいと思います。用意したプログラムはきっと皆さんの気に入っていただけると思います。ではまた。

リスト1

```
10 char doubt( 200, 1 ) /* 消せる可能性のあるタイル
20 int doubtEnd /* 疑ってみる最後のタイル
30 int clearEnd /* 消せる最後のタイル
40 int chkFlag /* まだ消せるかを表すフラグ
50 int points /* スコア
60 int levels( 19 ) = { /* レベル
70 +300, 600, 900, 1200, 1500,
80 +1800, 2100, 2400, 2700, 3000,
90 +3500, 4000, 4500, 5000, 5500,
100 +6000, 8000, 10000, 15000, 20000
110 }
120 int level=0 /* レベル保持用変数
130 /*
140 int tileX, tileY /* タイルの座標
150 int tmp /* タイルの番号
160 int tiles( 2 ) /* 落下中のタイル
170 int keySen /* キー入力フラグ
180 int gameOver
190 int i
200 str ch
210 /*
220 screen 1,3,1,1 /* 512x512x65536色
230 sp_init() /* スプライトの初期化
240 sprite_pattern() /* パターンの設定
250 sprite_color() /* パレットの変更
260 sp_disp( 1 ) /* スプライト表示
270 bg_set( 0, 1, 1 ) /* ベーグを表示
280 initFM() /* FM音源の設定
290 makeScreen() /* 画面作り
300 /*
310 locate 22,29 /* 乱数を
320 print "キーを押してください" /* 初期化する
330 while inkey$(0)="" /* ループ
340 rnd() /*
350 endwhile /*
360 locate 22,29 /* メッセージを
370 print chr$(5) /* 消す
380 while 1 /*
390 tileX = 16 /* 最初の位置
400 tileY = 3 /*
410 keySen = 1 /* キー入力可
420 /*
430 gameOver = 0 /* ゲーム続行
440 for i=0 to 2
450 if bg_get( 1, tileX, tileY+i ) <> 256 then {
460 gameOver = 1 /* 表示位置にタイルがあれば終わり
470 break
480 }
490 tiles( i ) = int( rnd()*6 + 2 ) * 256 + 1 /* 新しいタイル生成
500 bg_put( 1, tileX, tileY+i, tiles(i) )
510 next
520 if gameOver then break /* whileを抜ける
530 /*
540 m_play( 1 ) /* ウェイト開始
550 while inkey$(0)="" : endwhile /* 先行入力をクリア
560 while 1
570 if ( keySen ) then ch = inkey$(0) /* キー入力
580 switch ( asc(ch) )
590 case '4' /* 右へ移動
600 moveTile( tileX-1 ) : break
610 case '6' /* 左へ移動
620 moveTile( tileX+1 ) : break
630 case '5' /* タイル回転
640 tmp = tiles( 2 )
650 bg_put( 1, tileX, tileY+2, tiles( 1 ) )
660 tiles( 2 ) = tiles( 1 )
670 bg_put( 1, tileX, tileY+1, tiles( 0 ) )
680 tiles( 1 ) = tiles( 0 )
690 bg_put( 1, tileX, tileY, tmp )
700 tiles( 0 ) = tmp
710 break
720 case ' ' /* タイルを落とす
730 case '2'
740 keySen = 0
750 break
760 endwhile
770 if ch = chr$(27) then { /* 一時停止
780 while inkey$(0)="" : endwhile
790 while inkey$(0)="" : endwhile
800 }
810 }
820 /*
```

```
830 /* ウェイト終了、またはタイル落下の処理
840 /*
850 if m_stat( 1 )=0 or keySen=0 then {
860 if bg_get( 1, tileX, tileY+3 ) = 256 then {
870 for i=0 to 2 /* 下にタイルがなければ1つ下へ
880 bg_put( 1, tileX, tileY+i, 256 )
890 next
900 tileY = tileY + 1
910 for i=0 to 2
920 bg_put( 1, tileX, tileY+i, tiles(i) )
930 next
940 }
950 m_play( 1 ) /* 再びウェイトをかける
960 }
970 if bg_get( 1, tileX, tileY+3 ) <> 256 then break
980 endwhile
990 /* if gameOver then break
1000 for i=0 to 2 /* 3つのタイルをcheck配列に登録する
1010 doubt( i, 0 ) = tileX /* X座標を保存
1020 doubt( i, 1 ) = tileY+1 /* Y座標を保存
1030 next
1040 doubtEnd = 2 /* 疑ってみるのは0~2
1050 repeat /*
1060 check() /* 消せるかどうかチェックし
1070 chkFlag = erase() /* 消去する
1080 until chkFlag = 0 /* 消せるものなくなるまで
1090 endwhile
1100 end
1110 /*
1120 func moveTile( newx ) /* タイルの横への移動
1130 int i
1140 for i=0 to 2 /* 横に障害物はないか
1150 if bg_get( 1, newx, tileY+i ) <> 256 then return( )
1160 next
1170 for i=0 to 2 /* なければ移動
1180 bg_put( 1, tileX, tileY+i, tiles(i) )
1190 next
1200 for i=0 to 2
1210 bg_put( 1, newx, tileY+i, tiles(i) )
1220 next
1230 tileX = newx /* tileXを更新
1240 endfunc
1250 /*
1260 func makeScreen()
1270 int i
1280 fill( 0, 0, 511, 511, rgb( 0, 16, 0 ) )
1290 bg_fill( 1, 256 ) /* BGを256で埋める
1300 for i=5 to 26 /* レンガを縦に積む
1310 bg_put( 1, 11, i, 256*8+2 )
1320 bg_put( 1, 21, i, 256*8+2 )
1330 next
1340 for i=11 to 21 /* レンガの床を作る
1350 bg_put( 1, i, 26, 256*8+2 )
1360 next
1370 color 7 /* 白の太文字
1380 locate 25, 1
1390 print using "Points #####";0 /* 消したタイル数
1400 endfunc
1410 /*
1420 func initFM()
1430 int i /* 変数追加
1440 for i=1 to 30
1450 m_alloc( i, 40 ) /* トラックを30本確保
1460 next
1470 for i=1 to 20 /* 20本のトラックに
1480 m_trk( i, "@L"+str$( (20-i)*10+2)+"R" )
1490 next /* 次第に速くなる休符をセット
1500 /*
1510 m_assign( 1, 1 )
1520 m_tempo( 200 )
1530 m_trk( 25, "V15@5L16CCC" ) /* ポイント時のベルの音
1540 m_trk( 26, "V15@5L16EEF" ) /* ポイント時のベルの音
1550 m_trk( 27, "V15@5L16GAG" ) /* ポイント時のベルの音
1560 m_assign( 2, 25 ) /* ベルの音をセット
1570 m_assign( 3, 26 ) /* ベルの音をセット
1580 m_assign( 4, 27 ) /* ベルの音をセット
1590 endfunc
1600 /*
1610 func sprite_color()
1620 int i, j
1630 int r, g, b
1640 int cr, cg, cb /* Color of Red ... ..
```


PASCALの特徴的な性格について

Fujiki Takeshi Fujii Yoshimi
藤木健士/藤井義巳

PurePASCALは6月号で発表された X68000 用の PASCAL コンパイラです。PASCAL 言語の特徴的な概念について解説します。PurePASCAL の使い方は、6月号の付録ディスクの PASCAL.MAN というファイルを参照してください。

先月までのところでPASCALのだいたいの感じがわかったと思いますので今月はもう少しつっこんだ話をすることにしましょう。PASCALの特徴について述べてみたいと思います。FORTRAN全盛の時代に生まれたPASCALはそれまでになかった多くの考え方を導入しました。これによりプログラマは以前よりも安全性の高いプログラムを書けるようになり、以前は記述が困難であったアルゴリズムを記述できるようになったのです。それを可能にしたのは入れ子構造を反映したスコープ(有効範囲)、CALL BY VALUE、再帰呼び出しといったPASCALの持つ特徴です。これらのものは比較的新しいプログラミング言語においては常識となっているといっても過言ではありません。それでは今回はこれらの特徴に関して説明していきたいと思います。

スコープ

スコープとは定数、変数、関数および手続きの定義あるいは宣言された名前の見え方のことです。簡単な例を示してみましょう

```
1: PROGRAM TEST(INPUT, OUTPUT);
2:
3: VAR I, F: INTEGER;
4:
5: PROCEDURE FACT;
6:   VAR J: INTEGER;
7:
8:   BEGIN
9:     F:=1;
10:    FOR J:=1 TO I DO
11:      F:=F*J
12:    END;
13:
14: BEGIN
15:   WRITELN('INPUT I');
16:   READ(I);
17:   FACT;
18:   WRITELN('I=', I, ' FACT=', F)
19: END.
```

(注意: 行番号は説明のためにつけてありますが実際のプログラムにはありません)

簡単にプログラムの説明をしましょう。このプログラムは階乗を求めるプログラムです。このプログラムでは12行目までが宣言部で14行以降が本体になっています。それでこのプログラムは次のように動作します。

- 1) 'INPUT I'を表示する
- 2) Iをキーボードから読み込む
- 3) 階乗を計算する手続きをコールする
- 4) 結果を表示する

したがって、ステートメントの実行の順序は行番号でいうと15, 16, 9, 10, 11, 18という順になります。FACTという手続きはメインルーチンよりも上にありますが、これは宣言されているだけであって実際に実行されるのはそれが呼び出されたときです。それで実行の順序は上のようなのです。

このプログラムのようにPASCALでは定数、型、変数、手続きおよび関数は使用する前にすべて宣言されていなければなりません。それではWRITE, READ, INTEGERなどはどうなのでしょう。これらは事前に定義あるいは宣言されたものとして処理系が準備しているので宣言なしに使用してよいのです。

話がそれてしまいましたがスコープルールに話を戻しましょう。上でスコープルールとは名前の見え方に関する規則だといいましたが、次のように定義することができます。

名前はそれが定義または宣言されたブロックの中で有効である。

ブロックとは宣言部と本体から構成され、プログラム、手続きおよび関数はひとつのブロックから構成されます。上のプログラムの場合、1行目から19行目までがプログラムブロックで、5行目から12行目までが手続きFACTのブロックです。ということはこの手続きFACTは当然プログラムブロックに含まれていることになります。

それではこのプログラムの中で扱われている変数を例にとって具体的に説明することにしましょう。このプログラムの中では変数はI, Fがプログラムブロックで宣言されJがFACTという手続きのブロック中で宣言されています。IとFはプログラムブロックで有効ですから、プログラム中のどこからでも見えます(アクセスできる状態のことを「見える」とよくいいます)。それで手続きの中であっても9行目や11行目で変数Fにアクセスできるわけです。

このようにプログラムブロックで宣言された変数はプ

プログラム中のどこからでも見えますから、このことをグローバル変数（大域的な変数）と呼ぶこともあります。それに対してJは手続きの中で宣言されていますからこのブロックの中でしか使うことができません。

試しに14から19行目のあいだにJ:=1;というステートメントを入れてコンパイルしてみるとJが宣言されていないとエラーが出るでしょう。それでは次のような場合はどうなるのでしょうか。

```

1: PROGRAM TEST(INPUT, OUTPUT);
2:
3: VAR      I, F: INTEGER;
4:
5: FUNCTION FACT(X: INTEGER):INTEGER;
6:     VAR I, J: INTEGER;
7:
8:     BEGIN
9:         I:=1;
10:        FOR J:=1 TO X DO
11:            I:=I * J;
12:        FACT:=I
13:    END;
14:
15: BEGIN
16:     WRITELN('INPUT X');
17:     READ(I);
18:     F:=FACT(I);
19:     WRITELN('I=', I, ' FACT=', F)
20: END.
```

この例の中でIという変数が2回宣言されています(3行目と6行目)。さてこのプログラムはどのように動作するのでしょうか。この場合PASCALは内側のブロックで宣言された変数を有効にします。それで上のプログラムでは手続きFACTのブロックではプログラムブロックで宣言された変数Iは有効でなくFACTの中で宣言された変数が有効となります。つまり同じ名前が定義あるいは宣言されたときはその名前を使うブロックから見て一番内側のブロックで宣言されたものが見えるのです。6行目の変数Iの宣言をしないと、19行目で結果を書き出すときIの値がおかしくなるでしょう。それはFACTの中でプログラムブロックで宣言したIを更新してしまうからです。

この程度の短いプログラムだとなにかおかしいか探すのは容易ですが、長いプログラムだとそうもいきません。皆さんもこれに似た経験をしたことがありませんか？ 変わるはずのない変数の値が知らないうちに変わっててうまくプログラムが動かない！ 調べてみると実は別のところで使っている変数名を使ってしまっていたということはありませんか？ この手のバグは探すのが非常に困難です。

でもスコープがあるために、このような心配をする必要がなくなります。変数を使用するブロックで変数を宣言してしまえば、上位のブロックで同じ名前の変数があったとしてもそれが更新される心配をしなくていいと

いうことです。とにかく変数の見える範囲を選択できるのはプログラマにとって非常に有利な条件です。“変数はプログラムブロックで宣言すればいいや”なんて考えないでスコープを考慮して安全はプログラムを書くようにしましょう。ここでは変数を例にとって説明しましたが、定数、型、手続きおよび関数に関してもスコープは適用されます。

CALL BY VALUE

今度はCALL BY VALUEという考え方について説明しましょう。これは手続きや関数を呼び出すときに渡す引数に関する規則です。ここでも例を挙げて説明することにします。

```

1: PROGRAM TEST(INPUT, OUTPUT);
2:
3: VAR      I, F: INTEGER;
4:
5: FUNCTION FACT(X: INTEGER):INTEGER;
6:
7:     BEGIN
8:         FACT:=1;
9:         WHILE X > 1 DO
10:             BEGIN
11:                 FACT:=FACT * X;
12:                 X:=X-1
13:             END
14:         END;
15:
16: BEGIN
17:     WRITELN('INPUT I');
18:     READ(I);
19:     F:=FACT(I);
20:     WRITELN('I=', I, ' FACT=', F)
21: END.
```

これをFORTRANで書くと次のようになります。

```

1: PROGRAM TEST
2:   INTEGER I, F
3:
4:   WRITE(6, *) 'INPUT I'
5:   READ(5, *) I
6:   F=FACT(I)
7:   WRITE(6, *) 'I=', I, ' FACT=', F
8:   STOP
9:
10:  INTEGER FUNCTION FACT(X)
11:    INTEGER X
12:
13:    FACT=1
14:    DO 10 IF(X.GT.1) RETURN
15:      FACT=FACT*X
16:      X=X-1
```

```

17:      GOTO 10
18:
19:      END.

```

さてこれらはまったく同じ結果を出すような感じがしますが実際はそうではありません。結果からいうと、FORTRANで書かれたほうのプログラムでは7行目の表示の部分で必ず'I=1 FACT=XXXX'と表示してしまいます。Iが必ず1になるのです。さてこれはどうしてでしょうか？ PASCALとFORTRANの違いはどこにあるのでしょうか。

それは、FORTRANは変数のアドレスを渡し、PASCALは値を渡すという点です。上のPASCALの場合19行目で関数FACTに実引数としてIを渡していますが、これはIの値をFACTの仮引数Xにコピーして関数をコールしていると考えるとよいでしょう。それでPASCALのプログラムでは手続きの中でXの値を更新しても(12行目)Iの値は変わらないのです。

それに対してFORTRANのプログラムでは引数として変数のアドレスを渡すので、6行目のように実引数をIとしてFACTをコールした場合、Iのアドレスが引き渡されます。そして呼び出された関数FACTの中では、Xにアクセスするステートメントでは引数として与えられた変数のアドレスにアクセスします。例のプログラムでは6行目でFACT(I)というかたちで関数FACTの引数としてIを渡しているのです。FACT内の14行目15行目では変数Iを参照し、16行目では変数Iから1を引くのです。つまり変数Xの部分がすべてIに置き換えて処理をしたのと同じ結果を得るのです。それで16行目で1引くという処理は変数Iに対して行われ、結果としてI=1となってしまいます。

PASCALのように引数を値で渡すことをCALL BY VALUEと呼び、FORTRANのように引数をアドレスで渡すことをCALL BY REFERENCEと呼びます。CALL BY VALUEでは、関数や手続きを呼び出すときにパラメータとして渡した変数が更新されることがなくなります。これにより、より安全なプログラムを書くことができます。

ちなみにPASCALでもパラメータを値としてではなくアドレスとして渡すことができないわけではありません。たとえば上のPASCALのプログラムでパラメータをアドレスとして渡すようにしたいときは次のように書きます。

```

5: FUNCTION FACT(VAR X:INTEGER):INTEGER;

```

こうするとFORTRANのプログラムと同様の結果が得られるでしょう。複数の情報を呼び出した側に返すような関数を書きたいときなどにはこれを使うとよいでしょう。

再帰呼び出し(RECURSIVE CALL)

手続きや関数が自分自身を呼び出すことができるとアルゴリズムが記述しやすい場合があります。たとえばQUICK SORTなどはループで記述するよりも簡単に書く

ことができるのではないのでしょうか？ このように自分自身をコールすることを再帰呼び出しといいます。先の階乗を求めるプログラムを再帰呼び出しを用いて下のよう書くことができます。

```

1: PROGRAM TEST(INPUT, OUTPUT);
2:
3: VAR    I, F:INTEGER;
4:
5: FUNCTION FACT(I:INTEGER): INTEGER;
6: BEGIN
7:     IF I > 1 THEN
8:         FACT:=FACT(I-1)*I
9:     ELSE
10:        FACT:=1
11:    END;
12:
13: BEGIN
14:     WRITELN('INPUT I');
15:     READ(I);
16:     F:=FACT(I);
17:     WRITELN('I=',I,' FACT=', F)
18: END.

```

この9行目で関数FACTの中からFACTを呼び出しています。これは、

$N > 1$ のとき

$N! = N * (N-1)!$

$N = 1$ のとき

$N! = 1$

であることを記述しているのです。このように、数学の漸化式などの値を求めたいときには再帰呼び出しは便利です。最初は再帰のプログラムは難しいと感じますが、実際にこれを使ったほうが書きやすい場合がしばしばあります。読者の皆さんも使わず嫌いにならずに積極的にこれを使ってみることをおすすめします。再帰のプログラムは終了条件さえちゃんと書けばうまく動くものです。

この再帰呼び出しもFORTRANにない機能です。CALL BY VALUEと深い関係があり、それが実現されることによって簡単に再帰が可能になったのです。

最後に

今回はPASCALという言葉の持つ特徴の一部をそれ以前の言語と比較して紹介しました。今回説明した部分は私たちがPurePASCALを作るときにもけっこう悩んだ部分です。上で挙げた程度のものであればいいのですが、再帰したときのスコープはどうなるかと、2人で激論を戦わせたこともありました。結局わからないので大学の先生に電子メールを出して教えてもらったこともあります。これらに関しては非常に奥の深いものがあります。読者の皆さんもこのあたりのことに関心があるなら専門書を読んでみられてはいかがでしょうか。きっと得るものも多いと思います。

マシン語カクテル in Z80's Bar

第13回——電卓はビットシフトで——

シナリオ&イラスト：山田純二

特別監修：浦川博之 金子俊一



愛されて1周年

カランコロ〜ン

純二（以下純）：こんばんわ〜。おっ、みんな集まってますね。

古村（以下で）：あつ、不良の山田君。

純：（ドテッ）な、なんですかそれは……。

善司（以下善）：念願かなって買ったX6800が初期不良だったんでしょ。

純：い、いつの間に知れわたったんだ……。

マスター（以下M）：酒のサカナに私がいふらしたんですよ、はっはっは。

純：ひどいなあ、せっかくツケを払いにきてあげたっていうのに。

長老（以下老）：まあまあ、そんな話はひとまず置いて、こっちでパーッとやらんかの？

純：あれ、どうしたんです長老。手にメガホンなんか持ちちゃって。

老：こんなハズなメガホンがあるか。パーティだから盛り上げようとしとるんじゃないか。

純：へっ？ なんのパーティ？

ようこ（以下Yo）：あら、純二君気づいてないの？

で：もちろんショートプロバ……。

源光（以下光）：KATINOっ！

で：ぐー……。

老：実はじゃな、この7月号をもって、Z80's Barは、

純：最終回！

老：1周年じゃ、ばかもの！ いきなり縁起の悪いことをいうでない、まったく。（コホン）そういうわけで読者のみなさん、これからZ80's Barを、
一同：よろしくおねがいしまーす。

善：いえいえこちらこそ。

メアリー（以下メ）：Come on！ じゅん

じも、いっしょに騒グデース。

光：JAZZ'INもメッコールもあるよ。

純：……。

で：みんなー、長老のおごりだからどんどんいこー。

老：女の子の分だけだといったじやろうが。

善：そうよ、男の子はだめよ、ねえ長老。

老：南氷洋に沈めてほしいのか？

純：ねえ、なんか1周年のプレゼントとか記念品はないんですか？

M：山田君にはツケ払いに、プログラムの宿題をどっさり差しあげます。

純：（ちえっ）へーい。一応ひとつ作ってきたんですけど……。

メ：What kind of？

純：電卓……。

で：ただの？

老：まあそういうな。電卓にも四則演算などの重要なテクニックが入ってるんじゃないぞ。復習の意味もかねて見せてもらおうかの。



清く正しい乗除算

老：まずは乗除算のルーチンじゃな。

で：これは、足し算の繰り返しと引き算の繰り返しをしてるのかな？

光：確かに基本はそのとおりだけど、ちょっと情けないですね。

純：ははは……。

老：ビットシフトを使ったアルゴリズムぐらいはやってほしいところじゃな。

純：そういえば学校の授業でも聞いたことがあるような……。

M：どうせ、寝てるかマンガでも読んでたんでしょ。

純：睡眠学習といってください。馬鹿にしないでくださいよ、僕はこれで歴史を覚えたんですから。

善：鳴くよ（794）ウグイス、



なんだかんで、めでたく1周年を迎えてしまったZ80's Bar。1周年ということは13回目ということである。……不吉だ。しかしここに集まった面々にはそんなことはまったく関係なく、いつもどおり脳天気にお話を進めていくのであった。

純：平安京エイリアン。

光：筋金入りだね。

老：……とりあえず2進数8ビットの掛け算をやってみなさい。

純：10^{テン}-10（了解）。

$$\begin{array}{r} 10010110 \quad \cdots \cdots A \\ \times) \quad 00101001 \quad \cdots \cdots B \\ \hline 10010110 \\ 10010110 \\ 10010110 \\ \hline 1100000000110 \quad \cdots \cdots \text{答} \end{array}$$

となります。

老：よろしい。これを見てなにかわかるかな。

で：みんな1と0ばかりです。

光：だーかーらあー。

Yo：1のあるところだけ計算してるんじゃない？

老：そのとおり。

1) Bが1のビットを探す。

2) そのビットの桁にAを書く。

3) その桁のまま足す。

と、こういう順番になるな。

光：ちなみにAのほうを被乗数、Bのほうは乗数という。

善：人食いザメのことか。

で：それはジョーズでしょ。

善：あら、お上手。なんちて。

光：だあーつ、お前らはつ。

純：マシン語では、えーと、まずBを右に1ビットシフトする。キャリフラグでビットの0/1を調べて、0ならなにもしないで、1ならAを足す。それからAを左に1ビットシフトさせて、Bが0になるまで繰り返すんですね。

Yo：Aの左シフトってのはどうして？

純：10進数で見ればわかりますよ。

$$\begin{array}{r} 150 \\ \times) \quad 41 \\ \hline \end{array}$$

これは、まず 150×1 をやるでしょ。

Yo: ふんふん。

純: 次に41を右シフトすると4になるから 150×4 を計算する。だけど桁は右につまってるから、このままじゃ $150 + 600$ になっちゃう。だからあらかじめ600を左シフトして6000にしておけば、

$$\begin{array}{r} 150 \\ 6000 \leftarrow \text{筆算では書かない } 0 \\ \hline 6150 \end{array}$$

となるわけ。

Yo: 純二君すごーい。

純: いやあ、ははは。

光: 筆算どおりにやるならそのとおりでけど、ほかのやり方もあるんですよ、長老。

老: うむ、乗数(B)と答えを左シフトして求めるやり方じゃな。

純: へー、どうやるの?

光: まず、Bと答えを左に1ビットシフトさせる。キャリフラグでビットの0/1を調べ、0のときにはなにもせずにループを繰り返す。1のときには答えにAを足してやる。これを8回繰り返せばできあがり。

純: どう違うんですか?

老: 最初の方法は繰り返しが最大でも8回なのでスピードは速いが、プログラムが複雑なのじゃ。後者の方法は必ず8回繰り返すから若干遅いがプログラムは組みやすいな。

善: サンプルは後ろの方法だね。

老: どうじゃ、足し算だけでやるよりはエレガントじゃろう。

善: 確かにお鼻が長いなあ。

で: そりゃエレファントでしょうが。

* * *

M: 次は割り算ですね。

老: こちらのほうも、まずは筆算でやってみなさい。

純: らじゃ〜。

$$\begin{array}{r} 11 \cdots \cdots \text{商} \\ 101001 \overline{) 10010110 \leftarrow A} \\ \uparrow \quad 101001 \\ B \quad 1000100 \\ \quad 101001 \\ \hline \text{余り} \cdots \cdots 11011 \end{array}$$

(Aが被除数, Bが除数)

できましたよ。

老: よし。この筆算では、なにがどっちにシフトしながら計算を繰り返しているかわかるかな。

純: えーっとですね、1回計算するたびにBがだんだん右にシフトして、商は左にシフトしてます。

Yo: なんだかよくわかんないーい。

M: Oh! サツキトオナジョ。

純: だから、1回商が立つたびに、次の商を書くために商はひと桁上げてるってこと。これはさっきの掛け算と同じ理屈。で、除数のBのほうは書く位置が右にずれてるでしょ。要するに桁を小さくしてるわけ。

Yo: だって、Bを右シフトしたら数が変わっちゃうじゃない?

純: げっ……。

老: ほっほっほ。ようこちゃんのいうとおりじゃ。割り算はいちばん左の桁から計算を始めるんじゃないから、最初にBをAの1歩手前まで左シフトせんと計算が始められんな。

純: そうか、わかった。

- 1) 左シフトでBをスタート位置につける。
- 2) 商を左にシフトする。
- 3) Aがシフト前のBより小さくなってたら計算終了。
- 4) AからBを引いてみて、引けたら商を+1。だめだったらAを復元する。
- 5) Bを右シフトする。
- 6) 2へ戻る。

これなら100点でしょう?

老: うむ。実際のプログラムを見ると、このアルゴリズムとは一見違って見えるかもしれないが、基本の部分は同じじゃぞ。



変換しようよ!

純: ふー。なんとか乗除算はできた。

老: まだまだ先は長いぞ。電卓なら、打ち込まれた数字をコンピュータにわかる数値に変換せねばならん。

純: あ、それなら作ってきたこれで大丈夫だと思います。

老: フムフム……。純二君、ループの先頭でHLレジスタを10倍しているのはなぜなの?

純: 左端の桁から変換してるからです。

Yo: そんなこといったってわかんないーい。

純: つまり、10進数で3624という数は、 $3624 = 10^3 \times 3 + 10^2 \times 6 + 10^1 \times 2 + 10^0 \times 4$ というふうに表示することができるでしょ。それで、N桁目の数字は $10^{(N-1)}$ 倍されているというのをを使うわけ。1回目のループではHL=3で、次の桁に移ったときには、HLを10倍してHL=3×10+6になる。ループを繰り返すごとにHL=3×10²+6×10+2……最後にはHL=3624となるんだ。でしょ、長老。

老: 信ずる者は救われる、とは限らんが正解じゃ。それぞれの桁の重みを考慮することが大切なんじゃよ。

M: それじゃあ今度は、数値から数字への

変換ですね。

光: つまり16進数から10進数への変換なんだけど、純二君、一般の基数変換の授業は聞いてたかい?

純: 知ってる知ってる、これだけはさんざんやらされたんで、覚えてますよ。

光: よし、じゃあ説明してもらおう。

純: 基本は割り算の繰り返しです。変換したい数値を、基数で割っていったって、余りで下の桁から埋めていくんですよ。

老: なぜ下からか説明できるかな?

純: 割り算をするたびに商は1桁上のかたまりになるからかな? 3624割る10だと4が余って、362を10で割ると2が余って……という10進数→10進数の変換で見ればなんとなくわかると思いますけど。

老: 具体的には?

純: 3624 (10進数) を16進数に変換するには、

$$\begin{array}{r} 16) \underline{\quad} 3624 \quad \text{余り} \\ 16) \underline{\quad} 226 \quad \cdots \cdots 8 \\ \quad 14 \quad \cdots \cdots 2 \end{array}$$

14は16進でEだから、

$$3624(10進) = E28(16進)$$

というふうになります。

光: いちばんポピュラーな、人間向きのやり方ですね。

Yo: コンピュータ向きのやり方というのものもあるの?

光: コンピュータ相手には掛け算がネックなんです。加算、減算とビットシフトで処理したほうがコンピュータには楽なんです。

純: どうすればいいんですか?

老: それぞれの桁の重みを引いていけばよいのじゃ。

Yo: 1, 2, 4, 8, 16, 32ってやつ?

老: そのとおりじゃ。それが何回引けたかを、上の桁から順番に格納していけばよい。16ビットの16進数を10進の文字列にするときは、最大の桁数が5桁(65535)じゃから、5桁分の重みのテーブルを10000, 1000, 100, 10, 1と作ってやればよいな。

M: 納得できましたか?

純: できましたよ〜んだ。

Yo: 最初っから最後まで数字だったわね。頭が痛くなっちゃったわ。

光: 乗除算もしっかり説明しようとする結構大変ですね。

老: 情報数学専攻してる数学オンチもいるがなあ。みんなはここまでいわんでもわかると思うが。

で: ほっといてちょーだい。

M: 残りの部分はとするんですか?

純：メイン部分はまた来月払いということ
で……。

M：オヨヨ。今月の残りのサブルーチンの
説明ぐらいいはしてってくださいよ。

純：SEARCHSUBは演算子を判別させる
ための文字列判定ルーチン。そのほかのル
ーチンは見てのとおり、ワークからデータ
をPUSH,POPさせるものです。

Yo：あら？ そういえば善司君と（で）さ
んは？

M：さっきメアリーと帰っちゃいましたよ。

光：ぬわあに一つ（がたたん）！

Yo：（ギロツ）そんなに心配？

光：えっ、いやーはっはっは、そんなこ
とはぜーんぜん。じゃ、よーこさん、僕ら
も帰りましょうか。

Yo：私、お仕事残っているから。

光：そんなあ～。

老：さて、ワシらも帰るかのう。

純：へーい。さあさあ、光ちゃん帰りまし
よ。

M：ありがとうございました。来月こそツ
ケは完済してくださいよ。

カランコロヘン。



リスト1

```
0000 1
0000 2 :DENTAKU in Z80's Bar SUB
0000 3 : 1990.5.1 by Junji
0000 4
0000 5 #PRINT EQU $1FF4
0000 6 #PRINTS EQU $1FF1
0000 7 #LTNL EQU $1FEE
0000 8 #MSG EQU $1FE8
0000 9 #MSX EQU $1FE5
0000 10 #MPRINT EQU $1FE2
0000 11 #BELL EQU $1FC4
0000 12 #PRTHX EQU $1FC1
0000 13 #PRTHL EQU $1FBE
0000 14 #HLHEX EQU $1FB2
0000 15 #2HEX EQU $1FB5
0000 16 #FLGET EQU $2021
0000 17 #WIDCH EQU $2030
0000 18 #GETL EQU $1FD3
0000 19
0000 20 ORG $9000
0000 21
0000 22 JP ENTRY
0000 23
0000 24 :OUT ENZANSHI
0000 25 OUTENZ
0000 26 LD C,A
0000 27 LD B,$00
0000 28 LD A,$2
0000 29 CALL WPNWRITE
0000 30 RET
0000 31
0000 32 :PUSH DATA
0000 33 PUSH
0000 34 LD (PD+1),SP
0000 35 LD SP,(CALADR)
0000 36 PUSH BC
0000 37 LD (CALADR),SP
0000 38 PD LD SP,$0000
0000 39 RET
0000 40
0000 41 :POP DATA
0000 42 POPD
0000 43 LD (PD2+1),SP
0000 44 LD SP,(CALADR)
0000 45 POP BC
0000 46 LD (CALADR),SP
0000 47 PD2 LD SP,$0000
0000 48 RET
0000 49
0000 50 :ENZ PUSH
0000 51 ENZPUSH
0000 52 LD HL,(ESPADR)
0000 53 INC HL
0000 54 LD (HL),A
0000 55 LD (ESPADR),HL
0000 56 RET
0000 57
0000 58 ENZPOP
0000 59 LD HL,(ESPADR)
0000 60 LD A,(HL)
0000 61 DEC HL
0000 62 LD (ESPADR),HL
0000 63 RET
0000 64
0000 65 :WORK PUSH
0000 66 WPNWRITE
0000 67 LD HL,(WPADR)
0000 68 LD (HL),A
0000 69 INC HL
0000 70 LD (HL),B
0000 71 INC HL
0000 72 LD (HL),C
0000 73 INC HL
0000 74 LD (WPADR),HL
0000 75 RET
0000 76
0000 77 :WORK POP
0000 78 WPREAD
0000 79 LD HL,(WPADR)
0000 80 LD A,(HL)
0000 81 INC HL
0000 82 LD B,(HL)
0000 83 INC HL
0000 84 LD C,(HL)
0000 85 INC HL
0000 86 LD (WPADR),HL
0000 87 RET
```

```
905A 88
905A 89
905A 90 :文字列サーチ DE-文字列ポインタ
905A 91
905A 92 SEARCHSUB
905A 93 LD B,$00 :文字列番号
905A 94 SEA4
905C 95 LD (INSW),DE
9060 96 SEA3
9060 97 LD A,(DE)
9061 98 CP $20
9063 99 JR Z,SET? :文字列終了?
9065 100 CP $0D
9067 101 JR Z,SET?
9069 102
9069 103 LD C,A
906A 104 LD A,(HL)
906B 105 CP $FF
906D 106 JR Z,NOTERR
906F 107 OR A
9070 108 RET Z :文字列番号セット
9071 109 CP C
9072 110 JR NZ,SEA2
9074 111 INC HL
9075 112 INC DE
9076 113 JR SEA3
9078 114 SEA2
9078 115 CALL SKIPSUB
907B 116 SEA5
907B 117 INC HL
907C 118 INC B
907D 119 LD DE,(INSW)
9081 120 JR SEA4
9083 121 SET?
9083 122 LD A,(HL) :END CODE CHECK
9084 123 OR A
9085 124 JR NZ,SEA2
9087 125 OR A
9088 126 RET
9089 127 NOTERR
9089 128 SCF
908A 129 RET
908B 130
908B 131 :文字列スキップ
908B 132
908B 133 SKIPSUB
908B 134 LD A,(HL)
908C 135 OR A
908D 136 RET Z
908E 137 INC HL
908F 138 JR SKIPSUB
9091 139
9091 140 :加算 HL-HL+DE
9091 141 PLUS
9091 142 ADD HL,DE
9092 143 RET
9093 144
9093 145 :減算 HL-HL-DE
9093 146 MINUS
9093 147 OR A
9094 148 SBC HL,DE
9096 149 RET
9097 150
9097 151 :乗算 HL-HL*DE
9097 152 MULTI16
9097 153 LD C,E
9098 154 LD B,D
9099 155 LD DE,$0000
909C 156 LD A,16
909E 157 MUL3
909E 158 SRL B
909A 159 RR C
90A2 160 JR NC,MUL2
90A4 161 EX DE,HL
90A5 162 ADD HL,DE
90A6 163 EX DE,HL
90A7 164 MUL2
90A7 165 ADD HL,HL
90A8 166 DEC A
90A9 167 JR NZ,MUL3
90AB 168 EX DE,HL
90AC 169 RET
90AD 170
90AD 171 :除算 HL/DE,HL-ANS,DE-AWARI
90AD 172 DIV16
90AD 173 PUSH BC
90AE 174 LD A,16
```

```

90B0 4B      175      LD      C,E
90B1 42      176      LD      B,D
90B2 EB      177      EX      DE,HL
90B3 21 00 00 178      LD      HL,0000
90B6         179      DIV162
90B6 29      180      ADD     HL,HL
90B7 EB      181      EX      DE,HL
90B8 29      182      ADD     HL,HL
90B9 EB      183      EX      DE,HL
90BA 30 01   184      JR      NC,DIV163
90BC 23      185      INC     HL
90BD         186      DIV163
90BD B7      187      OR      A
90BE ED 42   188      SBC     HL,BC
90C0 30 03   189      JR      NC,DIV164
90C2 09      190      ADD     HL,BC
90C3 18 01   191      JR      DIV165
90C5         192      DIV164
90C5 13      193      INC     DE
90C6         194      DIV165
90C6 3D      195      DEC     A
90C7 20 ED   196      JR      NZ,DIV162
90C9 EB      197      EX      DE,HL
90CA C1      198      POP     BC
90CB C9      199      RET
90CC         200
90CC         201      MOD16
90CC CD AD 90 202      CALL    DIV16
90CF EB      203      EX      DE,HL
90D0 C9      204      RET
90D1         205
90D1         206      ; 1 O 進文字列を数値に変換
90D1         207
90D1         208      NUM10
90D1 ED 53 60 91 209      LD      (NUMWORK),DE
90D5 21 00 00 210      LD      HL,0000
90D8         211      NUM2
90D8 29      212      ADD     HL,HL
90D9 4D      213      LD      C,L
90DA 44      214      LD      B,H
90DB 29      215      ADD     HL,HL
90DC 29      216      ADD     HL,HL
90DD 09      217      ADD     HL,BC
90DE         218
90DE 1A      219      LD      A,(DE)
90DF CD 02 91 220      CALL    CHECK
90E2 38 0F   221      JR      C,NUMCHK
90E4 13      222      INC     DE
90E5 D6 30   223      SUB     0
90E7 4F      224      LD      C,A
90E8 06 00   225      LD      B,00
90EA 09      226      ADD     HL,BC
90EB 1A      227      LD      A,(DE)
90EC CD 02 91 228      CALL    CHECK
90EF 30 E7   229      JR      NC,NUM2
90F1 B7      230      OR      A
90F2 C9      231      RET
90F3         232      NUMCHK
90F3 E5      233      PUSH    HL
90F4 2A 60 91 234      LD      HL,(NUMWORK)
90F7 B7      235      OR      A
90F8 ED 52   236      SBC     HL,DE
90FA E1      237      POP     HL
90FB CA 00 91 238      JP      Z,NUMERR
90FE B7      239      OR      A
90FF C9      240      RET
9100         241      NUMERR
9100 37      242      SCF
9101 C9      243      RET
9102         244
9102         245      CHECK
9102 FE 3A   246      CP      97+1
9104 38 02   247      JR      C,CHECK2
9106         248      CHKEND
9106 37      249      SCF
9107 C9      250      RET
9108         251      CHECK2
9108 FE 30   252      CP      0
910A C9      253      RET
910B         254
910B         255      ;数値を1 O 進文字列に変換
910B         256
910B         257      STRING16
910B CD 22 91 258      CALL    STRSUB
910E 11 58 91 259      LD      DE,STRD16
9111 06 04   260      LD      B,04
9113         261      STR163
9113 1A      262      LD      A,(DE)
9114 FE 30   263      CP      0
9116 20 06   264      JR      NZ,STR162
9118 CD F1 1F 265      CALL    #PRINTS
911B 13      266      INC     DE
911C 10 F5   267      DJNZ    STR163
911E         268      STR162
911E CD E5 1F 269      CALL    #MSX
9121 C9      270      RET
9122         271
9122         272      STRSUB
9122 D9      273      EXX
9123 21 58 91 274      LD      HL,STRD16
9126 D9      275      EXX
9127 11 4E 91 276      LD      DE,DECTBL
912A 3E 05   277      LD      A,05
912C         278      STRLOOP
912C EB      279      EX      DE,HL
912D 4E      280      LD      C,(HL)
912E 23      281      INC     HL
912F 46      282      LD      B,(HL)
9130 23      283      INC     HL
9131 EB      284      EX      DE,HL
9132 08      285      EX      AF,AF
9133 97      286      SUB     A
9134 08      287      EX      AF,AF
9135         288      SLOOP2
9135 B7      289      OR      A

```

```

9136 ED 42   290      SBC     HL,BC
9138 38 05   291      JR      C,STRSUB2
913A 08      292      EX      AF,AF
913B 3C      293      INC     A
913C 08      294      EX      AF,AF
913D 18 F6   295      JR      SLOOP2
913F         296      STRSUB2
913F 09      297      ADD     HL,BC
9140 08      298      EX      AF,AF
9141 C6 30   299      ADD     A,0
9143 D9      300      EXX
9144 77      301      LD      (HL),A
9145 23      302      INC     HL
9146 D9      303      EXX
9147 08      304      EX      AF,AF
9148 3D      305      DEC     A
9149 20 E1   306      JR      NZ,STRLOOP
914B 97      307      SUB     A
914C 77      308      LD      (HL),A ;END CODE
914D C9      309      RET
914E         310
914E 10 27 E8 03 311      DECTBL DW 10000,1000,100,10,1
9152 64 00 0A 00
9156 01 00
9158 00 00 00 00 312      STRD16 DS 05
915C 00
915D 00
915E 00 00
9160 00 00
9162         313      DB 00
9162 2A 00   314      INSW  DW 0000
9164 2F 00   315      NUMWORK DW 0000
9166 44 00   316
9168 2B 00   317      CALTBL
916A 2D 00   318      DB 00
916C 28 00   319      DB 00
916E FF      320      DB 00
916F         321      DB 00
916F 97 90   322      DB 00
9171 AD 90   323      DB 00
9173 CC 90   324      DB 00
9175 91 90   325      DB 00
9177 93 90   326      JUMPTBL
9179         327      DW MULTI16
9179 00 00 00 00 328      DW DIV16
917D 00 00 00 00 329      DW MOD16
9181 00 00     330      DW PLUS
9183 00        331      DW MINUS
9184 04 92    332
9186 40 92    333      MEMDAT DS 10
9188 04 92
918A 00 00
918C 00 00 00 00
9190 00 00 00 00
9194 00 00 00 00
9198 00 00 00 00
919C 00 00 00 00
91A0 00 00 00 00
91A4 00 00 00 00
91A8 00 00 00 00
91AC 00 00 00 00
91B0 00 00 00 00
91B4 00 00 00 00
91B8 00 00 00 00
91BC 00 00 00 00
91C0 00 00 00 00
91C4 00 00 00 00
91C8 00 00 00 00
91CC 00 00 00 00
91D0 00 00 00 00
91D4 00 00 00 00
91D8 00 00 00 00
91DC 00 00 00 00 341      DS 40
91E0 00 00 00 00
91E4 00 00 00 00
91E8 00 00 00 00
91EC 00 00 00 00
91F0 00 00 00 00
91F4 00 00 00 00
91F8 00 00 00 00
91FC 00 00 00 00
9200 00 00 00 00
9204         342      CALSP
9204 00 00 00 00 343      WPSP DS 3*20
9208 00 00 00 00
920C 00 00 00 00
9210 00 00 00 00
9214 00 00 00 00
9218 00 00 00 00
921C 00 00 00 00
9220 00 00 00 00
9224 00 00 00 00
9228 00 00 00 00
922C 00 00 00 00
9230 00 00 00 00
9234 00 00 00 00
9238 00 00 00 00
923C 00 00 00 00
9240 07
9241 00 00 00 00 344      ENZSP DB $07
9245 00 00 00 00 345      DS 20
9249 00 00 00 00
924D 00 00 00 00
9251 00 00 00 00
9255
9255         346
9255         347      SUBEND
9255         348
9255         349
9255         350
9255         351
9255         352

```


X68000用

©日本テレネット

夢幻戦士ヴァリスIIより

SACRED SACRIFICE

Watanabe
Kazuhiko

渡辺 一彦

X1/turbo用

トッカータとフーガニ短調BWV565

Hanai Akitada

花井 章能

そろそろ巷では1学期期末試験などという、“恐怖のセレモニー”が待ち受けている頃でしょう。普段は“あの頃に戻りたい”などと言っている人も、この時期だけには戻りたくはありませんよね。ノートを借りまくっている人もいることでしょう。皆さん、いかがお過ごしでしょうか？ 私はコピー機を買いました。それから、先月号で予告していたT-SQUAREの“OMENS OF LOVE”は、ページの都合で来月回しとなっていました。申し訳ありません。自分の作品かな？ って期待していた人、本当にゴメンなさい。来月号まで待っていてください。

優子はいけにえ?

夢幻戦士ヴァリスIIより、オープニングテーマ「SACRED SACRIFICE」をお届けしましょう。直訳すると「聖なる生贄」と読めそうです。そうか、優子ちゃんは生贄だったのか……とひとりで納得していた私です。ちなみにX68000用のヴァリスIIではミュージックモードがあり、私服姿の優子ちゃんが見られます。ミュージックモードへの入り方は、オープニングデモの途中でF5キーを押してみてください。原曲と聴き比べるとわかりますが、渡辺君の作品はかなりスマートに仕上がっています。効果音バシバシ、ヴォリューム一杯で怒鳴っているような（失礼）あのオープニングテーマ

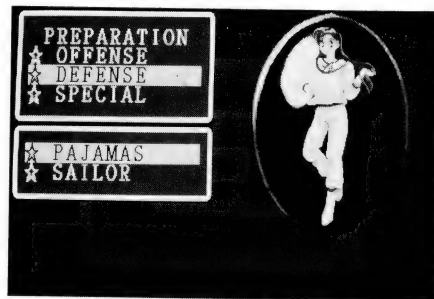
が、かなり洗練されています。騒ぐところは騒いで、抑えるところは抑えるという、メリハリが効いているのが素晴らしいでしょう。初投稿なのにこのレベルとは「なかなかやるわい」級といったところですね。ところで、原曲からしてヘビメタぽかったのに、作者の趣味に走ってしまって完全にそっちの世界に踏み外してるような気もしますが、大歓迎です。別にメタルじゃなかったっていいんです、完全に自分の世界に逃避してしまったようなアレンジバージョンも歓迎するということで、そちらのスジの方もお待ち申し上げております。

失われた惑星

X1用には「トッカータとフーガニ短調BWV565」をお届けしましょう。この曲はクラシックのなかでも、超ポピュラーな部類に入ると思えるくらい有名ですので、タイトルは知らない人でも、一度は聴いたことがあると思います。そういえば私の頼りない記憶では、幻魔大戦のサントラにも「失われた惑星」というタイトルで収録されたと思います。こちらのほうは、フルコーラスではなかったように記憶していますが……。この方面に詳しい人は、教えてください。

さて、作品の完成度なのですが、うーむ。全体的にはとってもよくできています。入力して損したと思うことはまずないでしょ

毎日雨様が続いていますが、皆さんいかがお過ごしでしょうか。さて、今月は予定を変更して、X68000用には夢幻戦士ヴァリスIIを、X1用にはトッカータとフーガをお送りします。しとしとと降るやさしい雨を眺めながら、たまにはじっくりとクラシックに浸ってみるのもいいもんですよ。



夢幻戦士ヴァリスII

う。エコーの残し具合や音の響き具合、間合いなどは苦心のあとがうかがえます。作者の花井君はよほど誰かの（演奏した）曲を聴きこんだのでしょう。かなり統一されたイメージで演奏されます。そういった面ですばらしかったからこそ、このページに載っていると思ってください。

ところで、問題点としては、部分的にブチノイズが残っていたりして、せつかくの曲の完成度を曇らせているところなのです。それ以外の出来のことを考えると、ちょっともったいないかな、って気がします。しょうがないと言ってしまうばそれまでののですが、やはりどうにかしたいというのが人心というものなのです。これは、音色の変更で軽減する可能性がありますので、各自研究してみてください。

LIVE inではこのページに対する希望やリクエスト、質問などもお待ちしております。アンケート葉書や、投稿作品に同封していただく手紙などに書いてくれたらうれしなっつ。（S.K.）

リスト1 夢幻戦士ヴァリスII

```
10 /*
20 /* The Fantasm Solder
30 /* 夢幻戦士ヴァリス II
40 /*
50 /*
60 /* OPENING Theme
70 /* SACRED SACRIFICE
80 /*
90 /* For OPMA 89/11/11 Sat. Ippiko. W
100 /* CD# アレンジ"ハ"ーシ"ヨンをもちに、かつてに"ふ"々々々7レンシ"をしようのよは"々々々々
110 m_init()
```

```
120 dim char v(4,10) : str m(99)[255],d(99)[99],sd1[10],sd2[10],sd3[10],bd[10],ht[10],mt[10],lt[10]
130 /*
140 for i=1 to 8
150 m_alloc(i,5000)
160 m_assign(i,i)
170 next : m_alloc(9,32)
180 /* Drums Set
190 sd1="y2,14
200 sd2="y2,17
210 sd3="y2,21
```

```

220 bd="y2,23
230 /*
240 ht="y2,24
250 mt="y2,25
260 lt="y2,26
270 /* Elec. Tom
280 d(0)=" (b&b-&a&a-&g&g-&f&e)8 "
290 d(1)=" (b&b-&a&a-&f&e)16
300 d(2)=" (b&b-&f&e)32
310 /*
320 d(10)=" (g&g-&f&e&e-&d&d-&c)8
330 d(11)=" (g&g-&f&e&e)16
340 d(12)=" (g&g-&f&e)32
350 /*
360 d(20)=" (e&e-&d&d-&c&c>b&b-&a)8
370 d(21)=" (e&e-&d&d-&c)16
380 d(22)=" (e&e-&f&e)32
390 /*
400 d(30)=" (c>b&b-&a&a-&g&g-&f)8
410 d(31)=" (c>b&b-&a&a)16
420 d(32)=" (c>b)32
430 /* MML Data
440 v={
450 /* A/F OM WF SY SP PMD AMD PMS AMS PAN nothing
460 56, 15, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0,
470 /* AR DR SR RR SL OL KS ML DT1 DT2 AME
480 17, 16, 0, 0, 3, 3, 26, 0, 3, 3, 0, 0,
490 20, 0, 0, 3, 0, 27, 0, 1, 0, 0, 0,
500 28, 0, 0, 3, 0, 28, 0, 1, 0, 0, 0,
510 28, 0, 0, 8, 0, 0, 0, 1, 4, 0, 0)
520 m_vset(70,v) /* Elec. Guitar
530 /*
540 v={
550 /* A/F OM WF SY SP PMD AMD PMS AMS PAN nothing
560 57, 15, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0,
570 /* AR DR SR RR SL OL KS ML DT1 DT2 AME
580 31, 11, 0, 2, 2, 36, 2, 9, 0, 0, 0,
590 31, 5, 0, 2, 1, 20, 1, 0, 0, 0, 0,
600 31, 10, 5, 5, 1, 34, 1, 0, 0, 0, 0,
610 31, 5, 2, 8, 1, 4, 1, 0, 0, 0, 0)
620 m_vset(71,v) /* Bass
630 /*
640 v={
650 /* A/F OM WF SY SP PMD AMD PMS AMS PAN nothing
660 59, 15, 2, 1, 0, 0, 0, 0, 0, 0, 3, 0,
670 /* AR DR SR RR SL OL KS ML DT1 DT2 AME
680 20, 0, 0, 2, 0, 15, 0, 7, 4, 0, 0,
690 31, 14, 0, 2, 2, 25, 0, 3, 0, 0, 0,
700 31, 12, 0, 2, 2, 20, 0, 4, 2, 0, 0,
710 31, 0, 0, 6, 0, 0, 0, 5, 7, 0, 0)
720 m_vset(72,v) /* Soft Dist. Guitar
730 /*
740 v={
750 /* A/F OM WF SY SP PMD AMD PMS AMS PAN nothing
760 62, 15, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0,
770 /* AR DR SR RR SL OL KS ML DT1 DT2 AME
780 31, 5, 5, 5, 2, 7, 0, 0, 1, 0, 0,
790 31, 16, 11, 8, 13, 0, 0, 0, 0, 0, 0,
800 31, 17, 10, 8, 12, 0, 0, 0, 0, 0, 0,
810 31, 12, 10, 8, 12, 0, 0, 0, 0, 0, 0)
820 m_vset(73,v) /* Tom Tom
830 /*
840 v={
850 /* A/F OM WF SY SP PMD AMD PMS AMS PAN nothing
860 60, 15, 2, 0, 0, 0, 0, 0, 0, 0, 3, 0,
870 /* AR DR SR RR SL OL KS ML DT1 DT2 AME
880 31, 10, 8, 15, 15, 0, 0, 12, 3, 1, 0,
890 31, 17, 10, 15, 15, 0, 0, 8, 3, 0, 0,
900 31, 10, 8, 15, 15, 0, 0, 12, 7, 1, 0,
910 31, 17, 10, 15, 15, 0, 0, 8, 3, 0, 0)
920 m_vset(68,v) /* Hi-Hat close
930 /*
940 v={
950 /* A/F OM WF SY SP PMD AMD PMS AMS PAN nothing
960 56, 15, 2, 0, 0, 0, 0, 0, 0, 0, 3, 0,
970 /* AR DR SR RR SL OL KS ML DT1 DT2 AME
980 26, 0, 8, 5, 3, 22, 0, 8, 4, 2, 0,
990 31, 11, 8, 7, 0, 30, 0, 8, 0, 2, 0,
1000 31, 11, 8, 7, 0, 25, 0, 8, 0, 1, 0,
1010 31, 17, 10, 7, 3, 0, 0, 4, 0, 0, 0)
1020 m_vset(82,v) /* Hi - Hat Open
1030 /*
1040 v={
1050 /* A/F OM WF SY SP PMD AMD PMS AMS PAN nothing
1060 59, 15, 2, 0, 200, 60, 0, 3, 0, 3, 0,
1070 /* AR DR SR RR SL OL KS ML DT1 DT2 AME
1080 31, 4, 2, 2, 15, 20, 0, 1, 4, 1, 0,
1090 31, 2, 0, 1, 15, 28, 0, 4, 0, 2, 0,
1100 31, 8, 0, 4, 15, 35, 0, 11, 0, 2, 0,
1110 31, 12, 4, 6, 15, 0, 0, 1, 3, 0, 0)
1120 m_vset(83,v) /* Ride Cymb
1130 /*
1140 m_tempo(132) /* Track Data
1150 /* track 1 (Main)
1160 m(0)=" r8 [do] y48,30 @70 p3 o4 v14
1170 m(1)=" l6 l: e-de-fe-d: d8r8d8c8&c1&c2
1180 m(2)=" (g&g&a&a&b&b&c&c&#d&e-&l)16&e-4..&e-1 l: {e-&e&f&f&g&g}16&g4&g16 l1 d1&d2&d8 l1 l2 c1&c8 {cde-fe-d}l
1190 m(3)=" c2..>b<c>116b16&b1
1200 m(4)=" c1 d1
1210 m(5)=" v15 l: p3 o4 g1 g1 f1 f1 e-1 e-1 d1 p1 g16g16r4g16g
16r4r4: l
1220 m(99)=" [loop]
1230 m_trk(1,m(0))
1240 m_trk(1,m(1))

```

```

1250 m_trk(1,m(1))
1260 m_trk(1,m(1))
1270 m_trk(1,m(1))
1280 m_trk(1,"o3"+m(2))
1290 m_trk(1,m(3))
1300 m_trk(1,m(2))
1310 m_trk(1,m(4))
1320 m_trk(1,m(5))
1330 m_trk(1,m(99))
1340 /* track 2 (Main ut")
1350 m(0)=" y48,10 r8 [do] @70 p3 o4 v12 r8
1360 m(4)=" c1 d2.. v12
1370 m(5)=" v15 l: p3 o4 g1 p1 r4e-2. p3 f1 p1 r4d2. p3 e-1 p1
r4c2. d1 < p2 g16g16r4g16g16r4r4 > : l
1380 m_trk(2,m(0))
1390 m_trk(2,m(1))
1400 m_trk(2,m(1))
1410 m_trk(2,m(1))
1420 m_trk(2,m(1))
1430 m_trk(2,"o3"+m(2))
1440 m_trk(2,m(3))
1450 m_trk(2,m(2))
1460 m_trk(2,m(4))
1470 m_trk(2,m(5))
1480 m_trk(2,m(99))
1490 /* track 2 (Sub)
1500 m(0)=" y50,30 r8 [do] @70 p1 o4 v13
1510 m(1)=" l6 l: o>b-<c cc>b-< : l >b-8r8b-8a-8&a-2&a-1 <
1520 m(4)=" c1 d1
1530 m(5)=" v15 l: p2 o4 r1 r2c2 > r1 r2b-2 r1 r2a-2 r1 p2 < g1
6g16r4g16g16r4r4: l
1540 m_trk(3,m(0))
1550 m_trk(3,m(1))
1560 m_trk(3,m(1))
1570 m_trk(3,m(1))
1580 m_trk(3,m(1))
1590 m_trk(3,"o3 v12 "+m(2))
1600 m_trk(3,m(3))
1610 m_trk(3,m(2))
1620 m_trk(3,m(4))
1630 m_trk(3,m(5))
1640 m_trk(3,m(99))
1650 /* track 2 (Sub ut")
1660 m(0)=" y51,10 r8 [do] @70 p2 o4 v11 r8
1670 m(4)=" c1 d2.. v12
1680 m(5)=" v15 l: p3 o3 r1 r2.a-4 r1 r2.g4 r1 r2.f4 r1 p1 < d1
6d16r4d16d16r4r4: l
1690 m_trk(4,m(0))
1700 m_trk(4,m(1))
1710 m_trk(4,m(1))
1720 m_trk(4,m(1))
1730 m_trk(4,m(1))
1740 m_trk(4,"o3 v12"+m(2))
1750 m_trk(4,m(3))
1760 m_trk(4,m(2))
1770 m_trk(4,m(4))
1780 m_trk(4,m(5))
1790 m_trk(4,m(99))
1800 /* track 2 (bass)
1810 m(0)=" @71 l8 v13 o2 y3,3 y2,23g16y2,23g16 [do] o3
1820 m(1)=" c2..> g16g16 <c1 > a-1&a-1 < l: 4 cc16c16c16c16c8 :
l > l: a-a-16a-16a-16a-16a- l: g16c->a-16<d>a16<d>-b16<c> (gab
-g)4 < l: l: 4 cc16c16c16c16c l: > l: 4 a-a-16a-16a-16a-16a- l: <
l: 4 cc16c16c16c16c l: > l: 4 a-a-16a-16a-16a-16a- l:
1830 m(2)=" < l: 4 c(cccc)4c l: > l: 4 b-(b-b-b-b)4b- l: l: 4 a-(
a-a-a-a)4a- l: l: g(gggg)4g l: g(gggg)4g g(gggg)4g <
1840 m(3)=" l: 4 c(cccc)4c l: > l: 4 b-(b-b-b-b)4b- l: l: 4 a-(a-
a-a-a)4a- l: l: 4 g(gggg)4g l:
1850 m(4)=" l: a-a-a-a-a-a-a-a- a-a-a-a-a-a-a-a- gggggggg gggggggg
ggg a-a-a-a-a-a-a-a- a-a-a-a-a-a-a-a- gggggggg g16g16r8 <g>g16g
16 r8<g8> (gab-g)4: l
1860 m_trk(5,m(0))
1870 m_trk(5,m(1))
1880 m_trk(5,m(2))
1890 m_trk(5,m(3))
1900 m_trk(5,m(4))
1910 m_trk(5,m(99))
1920 /* track 6 (Distortion Guitar)
1930 m(0)=" y53,60 @72 o2 v12 q7 g16g16 [do] o3 l8
1940 /m(2)=" < c1&c1 > b-1&b-1 a-1&a-1 g1&g1
1950 m(2)=" < l: 4 c(cccc)4c l: > l: 4 b-(b-b-b-b)4b- l: l: 4 a-(a-
a-a-a-a)4a- l: l: 4 g(gggg)4g l:
1960 m_trk(6,m(0))
1970 m_trk(6,m(1))
1980 m_trk(6,m(2))
1990 m_trk(6,m(2))
2000 m_trk(6,m(4))
2010 m_trk(6,m(99))
2020 /* track 7 (Distortion Guitar ut")
2030 m(0)=" y54,00 @72 o1 v12 q7 g16g16 [do] o2 l8
2040 m_trk(7,m(0))
2050 m_trk(7,m(1))
2060 m_trk(7,m(2))
2070 m_trk(7,m(2))
2080 m_trk(7,m(4))
2090 m_trk(7,m(99))
2100 /* Track 8 (Drums)
2110 m(0)=" r8 [do] y3,3 @v127 l8 o4
2120 m(1)=" y2,5r2..l: "+bd+"r16:l l: y3,3"+ht+"r48"+ht+"r12..y3
,1"+mt+"r48"+mt+"r12..y3,2"+lt+"r48"+lt+"r12..l: y3,3 l:12"+bd+"
r l: r4 y3,1 l: 4 y2,62 r32: l: y3,2 l: 4 y2,64 r32: l:
2130 m(2)=" y3,3 @83 p1"+bd+"b"+sd3+"b16&"+bd+"b16 b16&"+bd+"b1
6"+sd3+"b l:
2140 m(3)=bd+"r16"+"y2,24r64y2,24r16..y3,3"+bd+"r16"+"y3,1y2,25
r64y2,25r16..y3,3 "+bd+"r16 y3,2y2,26r64y2,26r16..y3,3"+bd+"r16

```



```

1240 "!"
1250 A="RRR4~L16>RFEDAR4>EDC+DL32E-DC<B-AG"
1260 B=A:C=A:D=A
1270 E="B->D<B-GFAFD<A>D<AFD>DC<B>C+B-AGFGFEDB-AGFGFEDL32EFGAB>
C+D16R8.<A16B>C+DEGL16AGFEF8R8"
1280 F=E:G=E:H=E
1290 "!"
1300 A="L8F+A4G>C<B>E-DE-<B>C<B>CDE-DE-FL32"+STRING$(33,"GF")+~L
160FGFE-FDE-CFE-FDE-CD<B->E-DE-CD<B->C<A>DCD<B->C<AB-GB-AB->C<B-A
G"
1310 B=A:C=A
1320 E="RE-DC<B-8B8>C8G8L32"+STRING$(25,"GF")+~L16GFGC-GDGCG<B>G
CGDGE-G<G>G<B>GCG<B>GCGDGE-8D8GA-FGE-8C8FGE-FD8<B-8>E-FDE-C8
F+8GAFGE-GF+GAGF+E"
1330 F=E:G=E
1340 I="L16O3RGFGE-GDGCG<B>GCGDGE-G<G>G<B>GCG<B>GCGDGL8E-DC<
B>CDE-FG<B>CDE-DE-FG<B>CDE-<AB->CD<GAB->CDGDE-<B-A>C"
1350 D="V10I2P1K5R8"+I
1360 H="V10I2P2K8R027"+I
1370 "!"
1380 A="F+8A8>D<G>C<F+B>G>D<AB-GAF+G>D<F+>D<G>D<A>D<B>G>D<AB-G>C
<A>DC<B-AB-AB-GAB->CDE-DCD<B->C<AB-G8B8>CE-FGA-GFGE-FDE-"
1390 B=A:C=A
1400 E="RDCD<B->D<A>D<G>D<F+>D<G>D<A>D<B>G>D<D>D<E>D<F+>D<G>D<F+>
D<G>D<A>D<B-8A8G2F+4G8A8B-AGFE-8>E-8D4C8D8"
1410 F=E:G=E
1420 D="D1&D1&D4R<G>C<A>D<DG4RG>C4.<B>C4"
1430 H=D
1440 "!"
1450 C="C8<B-8A8B-4A4G4F+8F8E-4DB-AB-AGF+>E-DC<B->AGF+G8.FE-FD"
1460 A=C+~E-"
1470 B=A
1480 D="V3I2P3K15R8L16O6"+C+~64"
1490 G="E-FDE-CE-DPGFE-FD<B>E-DC<A>DC<BG>C<B-AFB->DCDC<B-A>C<B
-AG>C<B-AB->C<B-AGAF"
1500 E=G+~G"
1510 F=E
1520 H="V3I2P3K15R021L16O5"+G+~64"
1530 "!"
1540 A="V9I2P1K5":C=A:E=A:G=A
1550 B="V9I2P2K8":D=B:F=B:H=B
1560 "!"
1570 A="O6L4C+DC<B-"
1580 B=A
1590 I="O5L16RB-AGA4F+4GFE-D"
1600 C=I+~R"
1610 D=I+~32."
1620 I="O5L16R4RE-DCDC<B-AG4R"
1630 E=I+~64"
1640 F=I+~09"
1650 G="O4L4EF+DG"
1660 H=G
1670 "!"
1680 A="V9I1P1K5":D=A
1690 B="V9I1P2K8":E=B
1700 C="V5I2P3K10":F=C
1710 G="V10I1P1K5"
1720 H="V8I2P3K8"
1730 "!"
1740 A="L8O5A>C+DEF4E2D4.FGDC+DEDC+DEDC+DEGFD<B->EDDC+4DEFEFCD+
+DEFEFGL16AB-GAFGEFD<A>C<A>D<A>E<A>F<A>E<A>F<A>G<A>"
1750 B=A:C=A
1760 D="O5C+8E8F8G8<D8>>D4.C+4L16<AB-GAL8FAGAGFGFEAB-AGFA>ED<AGG
AF+L16EAGAFADAC+ADAEFAFA<A>A<B>AC+ADAC+ADAEAL8FEDGFEFC+DC<D<B>"
1770 E=D:F=D
1780 G="L16O4AB-AGFGA>AGAFGAGAB-AGAFGEFDC<B-AB-AGFEFDC<A>D<A>
E<A>F<A>G<A>F<A>E<A>D<A>C<A>A>ADAF>D<G>D<E>C<F>D<D>D<L1A2&A&A&A
&"
1790 H=G
1800 "!"
1810 A="A<A>E<A>F<A>G<A>A4R1R2.REDC+D8RFEFGAB-AGAF8E8DEFE-DC<B-A
GAB->C<A4R8>C4<B-8R8>G8E-FDE-C<B-AGA4"
1820 B=A:C=A
1830 I="C+GFDL16C+<<~AGAFADAC+ADAEFAFA<A>A<B>AC+ADAC+ADAEFA>>RG
FEFC+D4.C+4DAB>C+D8C8<F4.E-8.C<AFA>CE-CD<B-GDGB>DGDC<B->C<AGF+E
F+DF+G"
1840 D=I+~R8"
1850 E=I+~09"
1860 F=I+~64"
1870 I="A&A&A2<~L8F4RFGEA<A>D4R<AB-G>C<CF4RAB-4RB->C4<RA>D4R"
1880 G=I+~16"
1890 H=I+~021"
1900 "!"
1910 A="V9I1P1K05":E=A
1920 B="V9I1P2K08":F=B
1930 C="V4I2P3K10":G=C
1940 D="V3I2P3K15":H=D
1950 "!"
1960 C="L16O4A>D<AG_F+DF+GA>D<AF+~GDGAB->D<B-A_GDGAB->DC<B-~AF+A
B->CE-C<B-~AF+AB->CE-C<B-~AB->DF+GB-GD<_B-GB->DGB-GDC<A>CE-F+AF+
E-C<A>CE-F+AF+"
1970 A=C+~E-"
1980 B=A
1990 D=C+~64"
2000 E="L4O3RDR"+STRING$(4,"~DR_DR")
2010 F=E
2020 G=E+~8."
2030 H=E+~027"
2040 "!"
2050 A="V10I2P1K5":C=A:E=A
2060 B="V10I2P2K8":D=B:F=B
2070 G="V12I2P1K5"
2080 H="V12I2P2K8"
2090 "!"
2100 A="L8O6R4DC<B-4>CDE-4FGA4FE-"

```

```

2110 B=A
2120 C="L8O5DRGF+G4GB>C4CEF4<B-A"
2130 D=C
2140 E="L8O4F+RGAB-4>E-<BG4>C<B-A4B->C"
2150 F=E
2160 G="L16O3RDCD<B->D<A>D<G>GFGE-GDGCC<B->C<A>C<G>C<F>FE-FDFCF"
2170 H=G
2180 "!"
2190 I="E-DC<B->C<B-AG>C<B-AGAF+GFEDC+8E8FGAB>C+DEFG8FEL8DC+DEF
GAC+DEFEDF"
2200 A="L16D"+I
2210 B=A
2220 E="B-8 V5I2P3K10L16">+I+~16"
2230 I="FE-DE-DC<B->E-DC<B->C<B-AB-AGFE8G8AB>C+DEFGAB-8AGFAEADAC
+ADAEFAFA<A>A<B>AC+ADAC+ADA<B>"
2240 C="L16DG"+I+~A"
2250 D=C
2260 F="B-8 V5I2P3K10R09L16">+I
2270 G="L8<B->E-F<FG>CD<DG4R16A16G16A16FDA4D1&D2>R4.D
2280 H=G
2290 "!"
2300 I="T70EDDC+T55DFE4.D"
2310 A=I+~2.R4."
2320 B=I+~0141R4."
2330 I="L8C+DGEFAB-.A.<F"
2340 C=I+~2.R4."
2350 D=I+~0141R4."
2360 E="V10P1K5L8<RAB-AA>DD.C.<F2.R4."
2370 F="V10P2K8R64L8RAB-AA>DD.C.<F0141R4."
2380 I="~AFGAFDG.A.B-"
2390 G=I+~2.R4."
2400 H=I+~0141R4."
2410 "!"
2420 A="T150V8I1P1K05":E=A
2430 B="V8I1P2K08 R64":F=B
2440 C="V4I2P3K10 R16":G=C
2450 D="V3I2P3K15 R08":H=D
2460 "!"
2470 A="L16O6F8GAB-<AB->CDCDE-F<FGAB-AB->CD<AGFE-GAB->C<GFE-DFGA
B-<AB->CDCDE-F<FGAB-AB->CDC<B-AGB->CDE-DC<B-A>CDEF+E-DC<B->F+T10
0GAB->C"
2480 B=A:C=A:D=A
2490 E="R1R1R1R1R0168"
2500 F=E:G=E:H=E
2510 "!"
2520 B="D.<G.> T70V10I2E2.D2.R1R1 T80V11D4C1&C1"
2530 A=B+~&C64"
2540 C="D.<G32 V10P1K5G2.F4D2R1R1 V11G+4A2.G8F8E1&E64"
2550 D="D015 V10P2K8<G2.F4D2O3L8R4G+.BFG+DF<B>D<G+A4R060 V11O5G+
A42.G8F8E1"
2560 F="O4G8. V10I2>C+2.D4<B2R1R1 V11G+4A1&G1"
2570 E=F+~&G64"
2580 G="O4G8 V10K5>E2.A4F2R1R1 V11<B4>C1&C1&C64"
2590 H="O4G16 V11K5<C+2F4.D8<G+2 T100L8>R4G+.BFG+DF<B>D<G+A4 V13
E1&E1&E011"
2600 "!"
2610 A="T150V8I1P1K05"
2620 B="V8I1P2K08 R64"
2630 C="V4I2P3K10 R16"
2640 D="V3I2P3K15 R08"
2650 E="":F=E:G=E:H=E
2660 "!"
2670 A="R1L16O5G.FGEEDCAGAFFEFDBABGGFGE>C<B>C<A>DCD<B>EDECFEFDG
FGEC<B>C<A>DCD<BG+G+E>C<B>C<AFEFDBABCBAGA<B>G+F+G+EC<BAG>+DC<BA
>EDC<B>FEDCGFEDAGFE8~G8A8B8"
2680 B=A:C=A:D=A
2690 E="L1RRRRRRRRR0138"
2700 F=E:G=E:H=E
2710 "!"
2720 A="T120V11I2P1K5 O6C1&C8"
2730 B="V11I2P2K8 O6C1&C8"
2740 C="V11I2P1K5 R010805E2"
2750 D="V11I2P2K8 R09905E2"
2760 E="V11I2P1K5 R012004G2"
2770 F="V11I2P2K8 R012304G2"
2780 G="V11I2P3K5 R012005G2"
2790 H="V12I2P3K5 R012003E2"
2800 "!"
2810 A="L16C+2.R4 T145<<A>C+EA>L8C+RC+RDRL16<<A>DFA>L8DRDR<BRL16
<GB>DGL8BRBR>CRL16<<G>CEG>L8CRCR<ARL16<FA>CFL8ARARB-RL16<FB->DFL
8B-RB-RB-RL16<EGB->C+L8B-RB-RARL16<DA>DF+A8R8"
2820 B=A
2830 C="E2.R4Q4L4R2EDR2FDR2DER2ECR2CDR2DER2C+DR2Q8"
2840 D=C
2850 E="A2.R4Q4L4R2AAR2AGR2GGR2GAR2FFR2FER2EF+R2Q8"
2860 F=E
2870 G="L16A2.R4RV5K10<A>C+EA>C+8RV11K5<A8R8F8R8.V5K10<A>DFA>D8R
V11K5<A8R8G8R8.V5K10<GB>DGB8RV11K5G8R8G8R8.V5K10<G>CEG>C8RV11K5<
G8R8F8R8.V5K10<FA>CFA8RV11K5F8R8F8R8.V5K10<FB->DFB-8RV11K5F8R8G8
R8.V5K10<EGB>+C+B-8RV11K5G8R8F+8R8.V5K10<DA>DF+A8RV11K5"
2880 H="G2.R4Q4L4R2GFR2FFR2FER2EER2EDR2DC+R2C+CR2Q8"
2890 "!"
2900 B="T100L1A2.GF2ET80D&D&D2.D."
2910 A=B+~&D64"
2920 D="L1D4<B->E2DC<B-2.A&A&A2"
2930 C=D+~&A64"
2940 F="L1F+4GB-2A2AF2G2.<F&F&F2"
2950 E=F+~&F64"
2960 G="L1F+4DC+2RRR2.<<D&D&D099"
2970 H="L2C4<GB-EFA1B-G.D1&D1&D099"
2980 "!"
2990 "
3000 PLAY "":;
3010 NEXT
3020 PLAY ""

```

▶あの事件以来ウィルスの怖さがわかった。このままでは人間不信（機械不信？）になりそう。
 大内 善雅（18） 福島県

前夜祭はすたつきいだ!

Komura Satoshi 古村 聡

やったー、今月は1周年だ。と思ったら勘違いだって……、とほほほ。まあいいか楽しみは先にとっておきましょう。そういうわけで今月は楽しい画面消去プログラム「ぱっくりあ」とパズルゲーム「STACKY」の2本です。

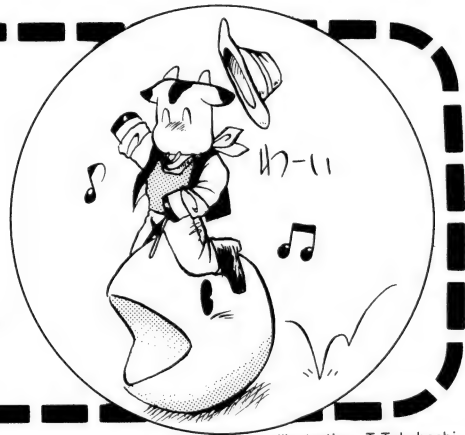
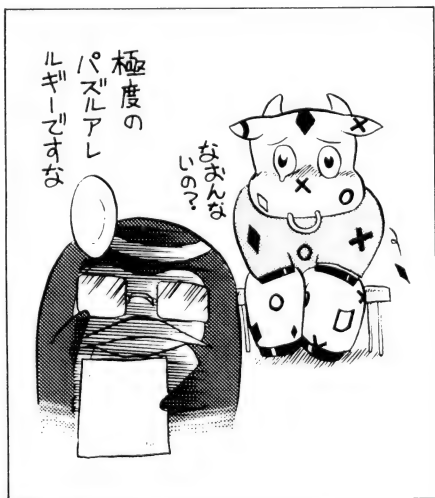


illustration: T. Takahashi

どもー。の一みそが1フレーズ足りないといわれる(で)です。

しょえーい、なんでこんなに凡ミスばかりするんだよお、わしって! そーです、読者の方からご指摘があったようにあのゼツパチマンの歌は1フレーズ書き落としていたのです。ああ、それだけならまだしも先月、「次はショートプロパてい1周年記念ぱてい」だとかいってたけど、よく考えてみたら1周年は来月じゃないか! ……だーっ、凡ミス大爆発! 1日や2日、日にちを間違えるのはまだしも(普通はそれだっけじゃないと思う)1カ月単位で間違えるなよー、何をやってんだかな本当に。

まあしょうがない、こうなったら今月は「次はショートプロパてい1周年記念ぱてい」だから、今夜はショートプロパてい1周年記念ぱてい前夜祭ぱてい」を行うぞ、だぜ!(冗談です、本当にごめんなさい。ささやかなショートプロ1周年記念ぱていは来月やります。しくしく。まあ、でも締め切りが1カ月延びたと思えばね……)



ぱっくりあぱっくりでCLS!

いやー、また今月もX68000のショートプロはマシン語だなー。ま、今月はマシン語特集だそうだし、いいか。

では、今月の1本目。今月の1本目は5月号掲載sp-chk()の豊田さんで、ふつーのCLSじゃつまんない! そんなふつーでないあなたのカスタムCLSプログラム、「ぱっくりあ.X」です。

ぱっくりあ.X for X68000

(command.x用)

神奈川県 豊田明彦

どういふものかっていうと早い話が画面をクリアするプログラムなんですけど、ただ画面をクリアするだけじゃあつまんない! んで、画面の文字をバックマンみたいなアニメーションがぱくぱくと消してってくれたりするんだな、これが。ちょっとアニメーションのパターンのドットが粗いような気もするんだけど(16×16ドットのキャラクターを無理やり32×32に伸ばして使ってるんだそーな)。ま、細かいことは気にしない気にしない。

で、例によって打ち込み方。今月はマシン語特集でそっちのほうでも書いてるんですけど(というより、書く予定なんだ、これから。あはは、せっかく提出した特集の原稿、総ボツくらってこれからぜーんぶ書き直しなんだよお……しくしく。ライターってつらい……)。とりあえずざっと説明しちゃいましょう。まず、

1) 先月号のディスクのおまけの

IOCSCALL.MAC

DOSCALL.MAC

をアセンブラ用に使っているディスクに

コピーする。

2) ED.Xを立ち上げる(コマンドラインから ED paccls.sと打つ)。

3) 載っているリスト打ち込む!

4) ESC [E] でED.Xを終わる。

5) AS PACLS と打ってアセンブル。

6) No Fatal error(s) と出なかったら2)に戻って間違えてるところを直す。

7) LK PACLS と打ってリンクすれば、……いっちゃあがり! です。

プログラムリストをじーっとながめればわかるけど、これって簡単にアニメパターンを描きかえられるうえに、アニメパターンが通ったあとの忘れ物まで描ける! つまりちよいちよいと手を加えれば「泥棒がものを落としながら走っていくCLS」なんてーのも簡単にできるんだな。ほら、最後のほうになんか0と1で絵が描いてある部分があるでしょ、そこをちよいちよいと手直してさっきの打ち込みと同じ手順でアセンブルしてやるだけなんです。投稿原稿にもせっかく「アニメパターンを4倍角にしたおかげでエディタで簡単にアニメパターンが描けますので気に入らない方は各自描き直してください」ってあることだし、こは存分に描き直させていただきます。私もおもわずイースIIIのクモ君にしてみたりしました(気持ち悪かった……)。

そうそう、投稿原稿で思い出したんですが「sp-chk()同様、このプログラムもフリーウェアにしたいと思います。ネットに流そうが改造しようが煮るなり焼くなりお好きにどうぞ」って書いてあったんですよ。ディスク集に間に合わなかったのがとても残念。電腦倶楽部にでも投稿しちゃおうか?



ワンパをねえ! スタッキーだ!

さて今月2本目のプログラムは2月号に掲載されたNUMBERSの坂本さんの投稿で、投稿原稿いわく(坂本さんの投稿原稿って好きなんだよねー、なんか楽しくって)「前作NUMBERSに続いてまたもやX1のグラフィック機能をまったく無視した低解像度ショートパズルゲーム第2弾「STACKY」(略称勝手に成りジャン、ちゃんくてスタッキー)」だそうです。ああ、つ、つかれる……。

STACKY for X1シリーズ

(CZ-8FB01用)

秋田県 坂本康

◎遊び方

ばらばらに並んでいるブロックがありますね。こいつを横1列に同じものが並ぶように並びかえるインテリジェンスた一つぶりのパズルゲーム(ひょえーい)だったりします。



こんなふうにね。

(ちょっとブロックの形が違うけど)

ただし一度に動かせるのは1つだけで上から順番に動かさなくてはけません。テンキーの4, 6で矢印をみぎひだりに動か

して選ぴスペースキーで決定します。ESCキーでメニュー画面になります。

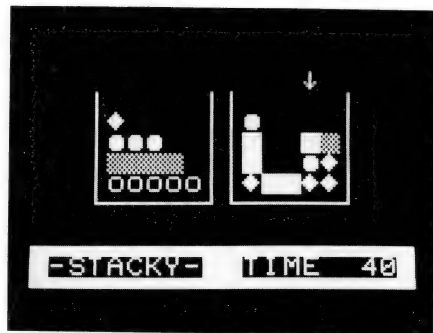
そうそう、メニュー画面では、TRY AGAIN……途中で詰まったときのやりなおし。

CORRECT ANSWER……どうしても解けないとき、解答の手順を見せてくれる。MODE CHANGE……ブロック数を3×3から8×8の範囲で選べてスペースキーで決定。

ということがそれぞれできるようになってます。

ようし、積んで積んで、また積んで……。あ、そっか、積んでいく様子がスタックを思わせるからスタッキーなわけか。納得。しかし、なんです、ね、「解答モード付きパズルゲーム」っていうのはいいもんです。自分が解けなかったらコンピュータがやっているとこながめてりやいいんだから。コンピュータのほうが人間さまより頭がいいというののもなんとなくやしいけど……。い、いか、どうせ1フレーズ抜けた男だもん(いじけってます、はっきりいって)。

さて、今月は今までショートプロに登場したおふたりさんの登場だったわけですが、そろそろ常連さんっていう人達も出てきそうですねー。なんか、私はまだよく見てないんだけど、投稿整理係さんの話だと「夜、見ないよーに」のデモの太田さんとかの新しい投稿とかももう届いてるって話だし。はたして誰が常連第1号と認定され



STACKY

るのか?

それにしても、やっぱり、得意分野があるのって強いですよ。とくにスタッキーの場合もあの4倍角文字のやつを作った人のですよっていっただけで、編集室でもなるほどってわかる人が多かったくらいですからね。世の中ワンパターンというとなにか悪いことのように聞こえるけど、私はそんなことって絶対ないと思います。だって、一流とか有名なブランドって必ずワンパターンじゃないですか。やっぱりワンパって人によく憶えてもらえるし、人にアピールするときに「私は〇〇で有名な××(名前)だ!」っていえるのって凄く強いと思うんですよ。それに特にプログラムを組む場合ってジャンルや言語を同じものでやってると技術の蓄積っていうのがありますから、上達が早いんですよ。

さあ、そういうことでみんなでワンパを目指せ! 常連への道は近いぞ! というわけでまた来月。

リスト1 ぱっくりあ

```
1: .list
2: *
3: *ぱっくりあ. X
4: * By Toyo
5: *
6: * 1990/05/14
7: .include DOSCALL.MAC
8: .include IOCSALL.MAC
9: .text
10: _txtvrm equ $e00000
11:
12: putank: lea usersp,sp
13: main: clr.b d3
14: clr.b d1
15: lopmn2: clr.b d0
16: lopmn: addq.b #1,d3
17: cmp.b #4,d3
18: boc prn1
19: prn0: lea fntdat,a0
20: bsr loopnai
21: bra nxtmn
22: prn1: lea fntdat2,a0
23: bsr loopnai
24: cmp.b #8,d3
25: bne nxtmn
26: clr.b d3
27: nxtmn: addq.b #1,d0
28: cmp.b #24,d0
29: bne lopmn
30: addq.b #1,d1
31: cmp.b #2,d1
32: bne lopmn2
33: cls: move.w #2,-(sp)
34: move.w #10,-(sp)
```

*IOCS・DOSコール使用

*テキストVRAMの先頭

*スタックの設定

*アニメーションのカウンタ

*Y方向のカウンタ

*X方向のカウンタ

*ハ・ターン1処理

*ハ・ターン2処理

*2の表示後、1の表示

*"next x"

*DOS上のCLS処理

```
35: DOS _CONCTRL
36: addq.l #4,sp
37: move.w #0,-(sp)
38: move.w #14,-(sp)
39: DOS _CONCTRL
40: addq.l #4,sp
41: DOS _EXIT
42:
43: loopnai: bsr putptn
44: bsr wait
45: bsr clrptn
46: rts
47: putptn: move.b #8,d4
48: mlop01: bsr putpl
49: add.b #2,d1
50: subq.b #1,d4
51: bne mlop01
52: sub.b #16,d1
53: rts
54: clrptn: lea fntdat3,a0
55: bsr putptn
56: rts
57:
58: putpl: movem.l d0-d6/a0-a6,-(sp)
59: bsr prn32
60: movem.l (sp)+,d0-d6/a0-a6
61: rts
62:
63: *wait: その名の通りウェイト
64: wait: movem.l d0-d6/a0-a6,-(sp)
65: move.l #100000,d5
66: waitl: nop
67: dbra d5,waitl
68: movem.l (sp)+,d0-d6/a0-a6
```

*作業終了

*上ルーフ内をサフ・ルーション化

*ハ・ターンのPUT

*7回分まとめてサフ・ルーション化

*PUTを使って画面クリアも

*細かいサフ・ルーション

*wait counter=140000

いよいよタマが撃てるぞ!

さあ、だんだんシューティングゲームっぽくなってきますよ。いよいよタマを撃ちます。にやー。なんちて。ぼくくん（これこれ、人の原稿に落書きをするなつて）。

ところで、ところで。タマを撃つにはどういふことが必要か？ ま、ジョイスティックのボタンを押すとタマが出る。ま、これはあたりまえなんです。たとえばシューティングゲームでタマが出てくるあいだ自機が動けなかったりしたらどうなるでしょう？ まず、ゲームにならないですよ。そう、ゲームっていうのは自分もタマも敵も一緒に動いてるんです。だから、今回はこういうプログラムの構造にします。

- 1) いま画面上にあるタマをさらに動かす
- 2) ジョイスティックのボタンが押されているかどうかを見る
- 3) ボタンが押されていてタマがまだ出せるようならタマを新しく出す

こんな感じのサブルーチン（関数と呼ぶのはなんか違和感があるな）を作ります。

で、ミソは 1) の「いま画面上にあるタマを動かす」の部分なんです。普通はコンピュータは一度にひとつのことしかできないものなんです。それでゲームにならない。そこで、一度にいろいろ動いてるように見せるために、

自機、タマ、自機、タマ……、

と順繰りに動かしてやるんです。だから自機を動かすルーチンにこのサブルーチンを呼び出す行を1行つけ加えてやれば、自機とタマが同時に動いてるように見えるつーわけです。ちなみに、敵と敵のタマが出てくると自機、タマ、敵、敵のタマ、自機、タマ……というふうに動かすわけです。わかりますよね。

そうそう、順繰りにって言えば、世の中にはワークステーションというえつらーいコンピュータがありまして、UNIXとかCとかが好きな人はおもわず「ははあつ」とひれ伏してしまうような（もちろん値段もむちゃくちゃ高い）ものだったりするんですね。で、これの特徴のひとつに「マルチタスク・マルチユーザー」という機能がありまして、ようするにこれって大勢の人がその1台のコンピュータでいっぺんにプログラムを動かすことができるんですね。そう、ワークステーションっていうのはまるでコンピュータの聖徳太子のようなありがたいものなんです。

……なんです。そういうコンピュータでも中身は実はそんなにX68000とかと基本的にはたいした差はないものなんですよ（といって、もちろん値段が高いたってスピードと

かは桁違いに凄いです）。実際、私の学校にもSUN3っていうありがたいワークステーションがあるんですけど、実はCPUは68000の兄貴分のMC68020っていうやつだしメモリに至っては4Mバイトしかないから、人によっては自分のX68000のメモリのほうが多いって人もいるんじゃない？ っていうぐらいのものなんです。でもそんな機械でなんでそんなことができるかっていうと、実はワークステーションの基本ソフト（これがUNIXというやつなの）がいろんなプログラムを順繰りにやるようにコンピュータを操ってただけだったりするんですよ。ま、正確にはひとつの作業ごとにやっているわけではなくて、タイムシェアリング(TSS)といってプログラムごとに一定の時間を割り振って実行していたりするんですけど。

ほかにもたとえばX68000でもゲームのキャラクターを動かしながらFM音源でじゃかじゃか音を鳴らしてたりするでしょ。あれも本当はキャラクターを動かしてキャラクターを動かして……ほい、時間だFM音源を鳴らして「ジャン!」、キャラクターを動かして、ほい、時間だFM音源を鳴らして「ジャジャン!」……っていうぐあいにちよっとずついろんなことをするようにしてるんですよ。

しかし、なんだな。なんか今回は関係ない話ばっかしとなる。でも、実は初心者なのにFM音源の話をするときに「FM音源って割り込みかけて一定間隔で作業をしてるだけで別にいっぺんにグラフィックと音楽をやってるわけじゃないんだよね。タイムシェアリングシステムっぽいって感覚かな」なんてさざっと言っておお、凄いととても尊敬されてしまいました、なんていう感謝のハガキが来るかもしれない。ま、よしとしよう。さらに「やっぱりマルチタスクはTSSだよね。イベントドリブンなんておもちゃ、おもちゃ」と言ってしまうとさらにグッドだな。そして、何かわからないことを質問されたら「やあ、最近SUNしか使ってなくて」とか言って逃げられれば完璧! かもしれない

(おいおい、これじゃショートプロバートいじゃなくて「初心者のためのパソコン見栄講座」じゃないか)。

ま、それはいいとして実際のプログラムを見ていきましょう。自機、タマを順繰りに動かすためには……。

まず、先月の160行から340行を見てください。自機の移動ルーチンになってますよね。その320行に、

```
firemove()
```

これです。あとはサブルーチンを作ればいいだけ、簡単でしょ。で、今回はゼビウスっぽくしようと思い、タマを1画面に3つ出せるようにしたかったんですよ。そこでタマの座標を、

```
fire_x(0~2)
```

```
fire_y(0~2)
```

という配列にしました。380行のfor文で3つのタマそれぞれの作業をするようにしています。……といってもスプライトを上になら動かすだけだから、

```
y=y-8
```

して、

```
sp_set(……)
```

すりゃ終わりなんだよね。

410行と450行に注釈が入ってますが、ここでタマが敵に当たったかどうかの判定をしているんですね。

あと、注意するところは390行でジョイスティックを調べる(strig(0))のがちょっと複雑になってるけど、これはジョイスティックが押しっぱなしになっているときにタマが連射にならないようにってんでつけたもんです。

とにかく、今回は座標を調べてループをかましてただけですからねー、そんなに難しくなかったでしょ？ それ以前に内容がなかったという話もあるけど、そんなことは気にしない気にしない。ね、編集担当者(いまごろ、これ読んで、怒りにぶるぶる震えてるだろうな)。

ま、今月はこんなところかな。来月またこのOh!Xで。シュワッ! (と空を飛んでくる)

```
320 firemove()
370 func firemove()
380 for b=0 to 2
390   trig=strig(1) and trig=k:strig(1)
400   if fire_x(b)>0 then (
410     /* atarif() 当たり判定が入る */
420     if fire_y(b)>0 then fire_y(b)=fire_y(b)-16
430     if fire_y(b)<16 then fire_x(b)=0 /*タマが上端(Y=0)の処理*/
440     sp_set(34+b,fire_x(b),fire_y(b),&H120)
450     /* atarif() */
460   ) else (
470     if trig=0 and k=1 then(
480       fire_x(b)=x:fire_y(b)=y-8:trig=1:sp_set(34+b,fire_x(b),fire_y(b),&H120)
490     )
500   )
510 next
520 endfunc
```

創刊8周年記念PRO-68Kのその後

編集部

先月号の付録ディスク「創刊8周年記念PRO-68K」はいかがでしたか。内容の密度はどこにも負けないつもりですが、いろいろと説明の不備があり、編集部には問い合わせが殺到しています。一応これまでにわかったことをまとめてみました。

ディスク解凍(展開)方法について

●Human68kのVer.1をご利用の場合

各プログラムを自動的に解凍するバッチ処理(DISK.BAT)はHuman68kVer.1では動作しないことがわかりました。残念ながらまとめて解凍するにはかえって面倒な処理が必要となります。申し訳ありませんが、6月号49ページの表1を見て必要なものから個別に解凍してください。

●Human68kのVer.2をご利用の場合

DISK.BATを使ってまとめて解凍する場合、SMPL(サンプリング音を集めたもの)が解凍されずに残ってしまいます。これを解凍するには、3枚目ディスクへの展開が終わった状態(A:に付録ディスク、B:にユーザーディスク)で、次のように打ち込んでください。

```
A>B:
B>RD SND
B>MD SMPL
B>CD SMPL
B>A:LH-E A:SMPL
```

これにより、3枚目のディスクにSMPLというディレクトリを作り、その中にサンプリングデータを展開することができるはずです。先月号49ページの解説には下線部のA:が抜けていました。申し訳ありません。

◆
Q: 付録のディスクを解凍しようとしたら「主記憶が不足しています」というエラーメッセージが出てしまいます。メモリを増設しなければ解凍できないのでしょうか。

A: ディスクを解凍するだけなら、増設の必要はありません。ただし、この春発売されたX68000PROIIシリーズの場合、システムディスクをそのまま使ってビジュアルシェルからコマンドモードを起動すると、展開のためのメモリが足りなくなってしまうことがわかりました。

メモリを確保するには、システムディスクをコマンドモードから立ち上げるように書き換えるとよいでしょう。システムディスクのCONFIGというディレクトリの中にあるTIT

LE.SYSをTITLE.VSのように変更(Rename)してください。次回からはコマンドモードから立ち上がるはずですが、詳しくは取扱説明書の第3部「3.4起動時にコマンドモードに入るには」(165ページ)をご覧ください。

各プログラムの利用について

付録ディスクにはシステムは入っていません。収録されているプログラムは単独では使えず、それぞれ必要に応じたシステムで起動しなければなりません。

また、X68000は機種によって付属のシステムディスクの内容が微妙に違いますし、ユーザーの使用状況によって対応のしかたも違ってきます。初心者の方にはわかりにくいことと思いますが、以下にだいたいの目安を示しますので参考にしてください。

●デバイスドライバの登録

基本的に、標準のシステムディスクは通常必要なほとんどのドライバが登録されていますが、機種によってその内容は異なる場合があります。たとえば、FM音源ドライバとして現在標準となっているOPMDRV.Xと浮動小数点演算パッケージFLOAT2.Xは、初代機種はシステムディスクには入っていませんでした。

OPMDRV.XとFLOAT2.Xはその後発売された数多くの市販ソフトに入っていますのでたいていの人はお持ちだと思いますが、これがないと、付録ディスクのYet Another Columnなども実行することができません。お持ちでない方はこの機会にHuman68kのVer.2.0を買うことをおすすめします。もっと安くする方法としては満開製作所発行のディスクマガジン電脳倶楽部を購読するという手もあります。事情が明記されたものに限り特別に1カ月(1,000円)だけでも応じてくれるそうです。

デバイスドライバを組み込むには、システムのなかにある、config.sysを書き換えます。一般にはED.Xというエディタを利用しますが、ここでは日本語ワードプロセッサWP.Xを使ってみましょう。

ワープロが起動したら、書き換えたいシス

テムディスクをAドライブに入れ、マウスで「ファイル入力」を選びます。ここでconfig.sysを読み込み、

```
DEVICE=A:¥OPMDRV.X
```

のように登録したいドライバのファイル名とその位置(この場合Aドライブのルートディレクトリ)を指定する1行を書き加えます。あとは「ファイル出力」を選んでconfig.sysを選択し、OKで上書きすれば終了。リセットで書き換えられたシステムが起動します。

逆に、ドライバを外す場合には、

```
*DEVICE=A:¥OPMDRV.X
```

のように*をつけるとそのドライバは無効になります。

●メインメモリの問題

メインメモリの空き容量は利用状況によって異なります。

たとえば、PROII/EXPERTIIに付属のシステムディスクは同じHuman68kのVer.2でも、BIOSを高速化するIOCS.Xが登録されているためシステムの使用メモリがそのぶん多くなっています。

このように、登録されているドライバによっても必要となるメモリが変わってきます。メモリに余裕のない場合はできるだけ不要なドライバを外してご使用ください。

●専用ディスクの作り方

各プログラムをシステムディスクに組み込んで使うにはディスクに十分な空き容量が必要です。お持ちの機種によってシステムディスクの空き容量はかなり違いますが、とりあえず、BINというディレクトリを削除すれば、500Kバイト以上の容量が確保できるはずです。

ウイルス検出プログラム

さて、X68000のウイルスですが、あなたのマシンとディスクは大丈夫だったでしょうか。問題のウイルスは、7月になると発病し大切なディスクのファイルを破壊する恐れがあります。まだ、チェックを行っていない方はぜひとも早めにチェックを行ってください。また、いまのところウイルスには冒されていないという方も予防のため、しばらくはDOCTOR.Rを組み込んでおくことをおすすめします。

Q: Oh!XのワクチンはSRAMを使うから危険だという話を聞きましたが本当に大丈夫なのでしょうか?

A: SRAM(スタティックRAM)を使うから危険ということはありません。また、X68000はSRAMがあるからウイルスに感染しやすいなどという人もいますがこれはナンセンスです。SRAMは比較的成本が高いので16Kバイトと小さな容量しかありませんが、ユーザーにも開放された便利な機能のひとつです。今回のウイルスはSRAMに常駐するタイプなので、ユーザーの皆さんが独自の用途に使っている場合、ウイルスの侵入でSRAMが書き換えられても即座に気づくことでしょう。

一般に、ワクチンは個別のウイルスに対してのみ有効で、どんなウイルスにも効く万能ワクチンというものはありません。しかし、SRAMにDOCTOR.Rを組み込んでおけば、少なくともIPL型のウイルスの侵入は防ぎやすくなりますし、SRAM常駐型のウイルスなら、万一侵入を許してもワクチン自体が破壊されるため発見が容易になります。

Oh!Xでは昨年6月号で発表したウイルスチェックをSRAMに組み込んで使っていました。そもそも、今回のウイルスが確認されたのはこのウイルスチェックが破壊されたためです。

今回のウイルスに手を加えた亜流が出てくる可能性も考えられますので、現在SRAMを使用していない方は当面DOCTOR.Rを組み込んだ状態で使用して下さるようご協力をお願いいたします。

Q: DOCTOR.Rを組み込んだのですが、ドライブになにもディスクを入れないで起動しようとすると、「IPLをチェックしています。セルフチェックOK」「ドライブ0にHumanのディスクを入れてください」のあと「エラーが発生しました。リセットしてください」と出るのです。なぜでしょう。私のX68000がおかしいのでしょうか?

A: 心配ありません。というよりエラーとなるのは正常な証拠です。DOCTOR.Rを組み込んでおくと、電源ON(またはリセット)でシステムが起動する前にそのディスクがウイルスに冒されていないかチェックするようになっています。しかし、ディスクが準備できていない場合、ディスク挿入と同時に起動させてしまうとウイルスの侵入を許すことになってしまいます。そこで、もう一度リセットをかけてディスクをチェックするようわざとエラーを発生させているのです。

ただし、ふつうのハードディスクをつないでいる場合、ディスクが準備されていなければ素直にハードディスクから起動するため、エラーとはなりませんし、感染の恐れもありません。また、SCSIタイプのハードディスクを使っている場合は起動をSCSIデバイス

1にしないと立ち上がらないようなので、DOCTORはコマンドラインからの使用のみでウイルスチェックをしなければなりません(TF R2は使わないでください)。

また、SRAMを初期状態に戻すには、

A>SWITCH B=ROM0

A>SWITCH B=STD

と打ち込んでください。

Yet Another Column

このゲームはBASICで作成されたものをコンパイルしたものです。プレイするには、FM音源ドライバOPMDRV.Xと浮動小数点パッケージ“float2.x”(または“float2+”)を組み込んだシステムディスクを使用してください。また、実行ファイルは“yet.x”, 背景のグラフィックデータは“siro.gs3”です。

とりあえずゲームを楽しみたい方は、BINを削除したシステムディスクをAドライブで立ち上げ、Bドライブに付録ディスクを入れます。コマンドモードで、

A>B:LH-E B:GAMES

としてください。展開が終了したら、

A>YET

と打ち込めばゲームが始まります。以後はビジュアルシェルスでyet.xのアイコンをダブルクリックしてもOKです。

音楽演奏関係

Q: ナイトアームズのエンディングテーマの演奏方法がよくわかりません。

A: 53ページ右下のOPMDサンプル曲の説明にはいろいろと不備がありました。

単独でOPMDとSMPLを解凍する場合、まず、BASICの入った500Kバイト以上空きのあるシステムディスクをAドライブ、オマケディスクをBドライブに入れ、コマンドモードから、

A>B:LH-E B:OPMD

A>B:LH-E B:SMPL

としてください。これは先月の囲み記事のとおりです。

また、DISK.BATでまとめて解凍した方は、SMPLを単独で解凍したあと、用意したシステムディスクにOPMDとSMPLの中身をすべてコピーしてください。システムを作成するディスクをAドライブ、3枚目の解凍ディスクをBドライブに入れ、

A>COPY B:OPMD

A>COPY B:SMPL

と入力します。

内蔵音源のみの場合なら、

A>OPMD /A BOS

MIDIでM1を鳴らすなら、

A>OPMD /M KAEND

同じくMT-32の場合、

A>OPMD /M KEMT

として、OPMD.Xを立ち上げます。

あとは、

A>BASIC

でBASICを立ち上げ、

LOAD“KAEND.BAS”

RUN

でOKです。

Q: MUSICDRVで設定した音色と変更された音色が違うのですが……。

A: MUSIC2.FNCを使うと音色登録時に番号がひとつずれてしまいます。以下の修正を加えてください。まず、

A>MAC

としてTOOLSに入っていたマシン語入力ツールを起動します。NEW FILEかどうかにはNと答えて、ファイル名にMUSIC2.FNCを指定してください。

プログラムが読み込まれたら、“E”でエディットモードに入り、

0451 53 → 4E

0452 80 → 71

0457 80 → 81

0495 53 → 4E

0496 80 → 71

049C 80 → 81

のようにデータを書き換えてください。この6カ所を変えればあとはSキーを押してセーブし「!」で処理を中止します。

また、MUSIC.DRVはOPMDRVと同時に使えません。

そのほかのプログラムについて

Q: PIC.FNCを組み込むとBASIC起動時にエラーが出るのですが。

A: ファイル内のデータが2バイトずれていました。原因はよくわかりません。とりあえず、PIC.FNCをPICO.FNCにリネームしたうえでBASICからリスト1を入力、実行してください。また、コンパイルして使う場合は、各関数のパラメータは省略しないようにしてください。

リスト1

```
10 /* *****PIC.FNC debug
20 /* freopen("pic.fnc","pic0.fnc")
30 char a(5000)
40 int i,j,k
50 /*
60 i=fopen("pic0.fnc","r")
70 j=fopen("pic.fnc","c")
80 fread(a,&H400,i)
90 fwrite(a,&H400,j)
100 fputc(&H2F,j):fputc(&H3C,j)
110 fread(a,4724-&H400,i)
120 /*
130 for k=0 to 36
140   a(k*4+&H16F)=a(k*4+&H16F)+2
150 next
160 a(&HE17)=a(&HE17)+2
170 fwrite(a,4724-&H400,j)
180 fcloseall()
```

Q:もしかしてPurePASCALにはアセンブラが必要なのでしょうか。

A:特に触れていませんでしたが、PurePASCALのコンパイラはアセンブラのソースを出力する形式をとっています。したがって、実行ファイルの作成にはアセンブラおよびリンカ (AS.X, LK.Xなど) が必要です。PurePASCALのシステムディスクを作成する場合はこれらを忘れずに組み込んでください、なお、コンパイルスイッチをなにも指定しなければ、*.s, *.o, *.xの順で実行ファイルを自動的に作成するようになっていきます。詳しくはドキュメントPASCAL.MANをご覧ください。

Q: PurePASCALでFLOAT2+/3+が正常に動きません。

A: 現在調査中です。FLOAT2+/3+は使わないでください。

Q: ディスクに入っていた99.BASはどうやって起動すればいいのですか。

A: 99.BASはBASIC用CARD.FNCのサンプルです。CARD.FNCの組み込み方がわからない人は以下にその1例を書きましたのでこのとおりに実行してみてください。

1) まず、システムディスクをコピーします (AドライブからBドライブへ)。

注) システムディスクの空き容量が135Kバイト以上ない場合はコピーしたあととコマンドモード上で、

```
DEL B:BIN
```

```
COPY A:¥BIN¥ED.X B:
```

としてBINを殺してディスク容量を確保します。

2) コピーしてできたディスクをAドライブに、オマケを展開したディスク (GAMESというディレクトリが入っているもの) をBドライブに入れます。

3) BASICと打ち込んでBASICを起動します。

4) 以下のコマンドを実行してください。

```
CHDRV"B:"
```

```
CHDIR"GAMES"
```

```
LOAD"MAKE.BAS"
```

```
RUN
```

これでデータが展開されて、CARD.FNCというファイルが出来上がります。

5) 次に、SYSTEMと打ち込みコマンドモードに戻ります。

6) ここで、

```
COPY B:¥GAMES¥CARD.FNC A:BASIC2
```

(初代X68000についてきたシステムディスクの場合はBASIC2のところをBASICにする) として、BASICのディレクトリにCARD.FNCをコピーする。

7) コマンドモードで

```
A>ED ¥BASIC2¥BASIC.CNF
```

として、エディタを起動します (さっきと同

じように、BASIC2じゃない場合はBASICに変える)。

8) 画面にBASIC.CNFの内容が表示されますので、そこに、

```
FUNC=CARD
```

と1行書き加えてください。

9) ESC+Eを押してセーブします。

以上でCARD.FNCのBASICへの組み込みは完了です。

Q: DIS.Xの解説にあったFEFUNC.Hはディスクには入っていませんがどうすればいいのでしょうか。

リスト2

```
===== FEFUNC.H =====
1: .nlist
2: *
3: * fefunc.h X68k XC Compiler v1.01 Copyright 1987 SHARP/Hudson
4: *
5: FPACK      macro    callname
6:   dc.w      callname
7:   endm
8:
9: *****
10:  __LMUL      EQU      $FE00      d0=d0*d1
11:  __LDIV      EQU      $FE01      d0=d0/d1
12:  __LMOD      EQU      $FE02      d0=d0 mod d1
13:  *
14:  __UMUL      EQU      $FE04      d0=d0*d1
15:  __UDIV      EQU      $FE05      d0=d0/d1
16:  __UMOD      EQU      $FE06      d0=d0 mod d1
17:  *
18:  __IMUL      EQU      $FE08      d0d1=d0*d1
19:  __IDIV      EQU      $FE09      d0777d1=d0/d1
20:  *
21:  *
22:  __RANDOMIZE  EQU      $FE0C      err=d0(0~65535)
23:  __SRAND     EQU      $FE0D      err=d0(-32768~32767)
24:  __RAND      EQU      $FE0E      d0=rand()
25:  *
26:  *****
27:  __STOL      EQU      $FE10      d0=(a0).1 0 進
28:  __LTOS      EQU      $FE11      (a0)..=d0
29:  __STOH      EQU      $FE12      d0=(a0).1 6 進
30:  __HTOS      EQU      $FE13      (a0)..=d0
31:  __STOO      EQU      $FE14      d0=(a0).8 進
32:  __OTOS      EQU      $FE15      (a0)..=d0
33:  __STOB      EQU      $FE16      d0=(a0).2 進
34:  __BTOS      EQU      $FE17      (a0)..=d0
35:  __IUSING    EQU      $FE18      (a0)..=d0,d1桁
36:  *
37:  *****
38:  __LTOD      EQU      $FE1A      d0d1=d0
39:  __DTOL      EQU      $FE1B      d0=d0d1
40:  __LTOF      EQU      $FE1C      d0=d0
41:  __FTOL      EQU      $FE1D      d0=d0
42:  __FTOD      EQU      $FE1E      d0d1=d0
43:  __DTOF      EQU      $FE1F      d0=d0d1
44:  *****
45:  __VAL       EQU      $FE20      d0d1=(a0).16/10/8/2 進
46:  __USING     EQU      $FE21      (a0)..=d0d1,d2.d3,d4
47:  __STOD      EQU      $FE22      d0d1,d2,d3=(a0).1 0 進
48:  __DTOS      EQU      $FE23      (a0)..=d0d1
49:  __ECVT      EQU      $FE24      (a0),d0,d1=d0d1,d2 全体桁
50:  __FCVT      EQU      $FE25      (a0),d0,d1=d0d1,d2 小数点桁
51:  __GCVT      EQU      $FE26      (a0)..=d0d1,d2 全体桁
52:  *
53:  *****
54:  __DTST      EQU      $FE28      z=d0d1
55:  __DCMP      EQU      $FE29      z,c=d0d1-d2d3
56:  __DNEG      EQU      $FE2A      d0d1=neg(d0d1)
57:  __DADD      EQU      $FE2B      d0d1=d0d1+d2d3
58:  __DSUB      EQU      $FE2C      d0d1=d0d1-d2d3
59:  __DMUL      EQU      $FE2D      d0d1=d0d1*d2d3
60:  __DDIV      EQU      $FE2E      d0d1=d0d1/d2d3
61:  __DMOD      EQU      $FE2F      d0d1=d0d1 mod d2d3
62:  *****
63:  __DABS      EQU      $FE30      d0d1=abs(d0d1)
64:  __DCEIL     EQU      $FE31      d0d1=ceil(d0d1)
65:  __DFIX      EQU      $FE32      d0d1=fix(d0d1)
66:  __DFLOOR    EQU      $FE33      d0d1=floor(d0d1)
67:  __DFRAC     EQU      $FE34      d0d1=frac(d0d1)
68:  __DSGN      EQU      $FE35      d0d1=sgn(d0d1)
69:  __DSIN      EQU      $FE36      d0d1=sin(d0d1)
70:  __DCOS      EQU      $FE37      d0d1=cos(d0d1)
71:  __DTAN      EQU      $FE38      d0d1=tan(d0d1)
72:  __DATAN     EQU      $FE39      d0d1=atan(d0d1)
73:  __DLOG      EQU      $FE3A      d0d1=ln(d0d1)
74:  __DEXP      EQU      $FE3B      d0d1=exp(d0d1)
75:  __DSQR      EQU      $FE3C      d0d1=sqr(d0d1)
76:  __DPI       EQU      $FE3D      d0d1=pi()
77:  __DNPI      EQU      $FE3E      d0d1=pi(d0d1)
78:  __DPOWER    EQU      $FE3F      d0d1=pow(d0d1,d2d3)
79:  __DRND      EQU      $FE40      d0d1=rnd()
```


80: *	EQU	\$FE41	
81: *	EQU	\$FE42	
82: *	EQU	\$FE43	
83: *	EQU	\$FE44	
84: *	EQU	\$FE45	
85: *	EQU	\$FE46	
86: *	EQU	\$FE47	
87: *	EQU	\$FE48	
88: _DFREXP	EQU	\$FE49	D0D1,D2=D0D1
89: _DLDEXP	EQU	\$FE4A	D0D1=D0D1,D2
90: _DADDONE	EQU	\$FE4B	d0d1=d0d1+1#
91: _DSUBONE	EQU	\$FE4C	d0d1=d0d1-1#
92: _DDIVTWO	EQU	\$FE4D	d0d1=d0d1/2#
93: _DIEECNV	EQU	\$FE4E	d0d1=d0d1
94: _IEEDCNV	EQU	\$FE4F	d0d1=d0d1
95: *****			
96: _FVAL	EQU	\$FE50	
97: _FUSING	EQU	\$FE51	
98: _STOF	EQU	\$FE52	
99: _FTOS	EQU	\$FE53	
100: _FECVT	EQU	\$FE54	
101: _FFCVT	EQU	\$FE55	
102: _FGCVT	EQU	\$FE56	
103: *	EQU	\$FE57	
104: *****			
105: _FTST	EQU	\$FE58	
106: _FCMP	EQU	\$FE59	
107: _FNEG	EQU	\$FE5A	
108: _FADD	EQU	\$FE5B	
109: _FSUB	EQU	\$FE5C	
110: _FMUL	EQU	\$FE5D	
111: _FDIV	EQU	\$FE5E	
112: _FMOD	EQU	\$FE5F	
113: *****			
114: _FABS	EQU	\$FE60	
115: _FCEIL	EQU	\$FE61	
116: _FFIX	EQU	\$FE62	
117: _FFLOOR	EQU	\$FE63	
118: _FFRAC	EQU	\$FE64	
119: _FSGN	EQU	\$FE65	
120: _FSIN	EQU	\$FE66	
121: _FCOS	EQU	\$FE67	
122: _FTAN	EQU	\$FE68	
123: _FATAN	EQU	\$FE69	
124: _FLOG	EQU	\$FE6A	
125: _FEXP	EQU	\$FE6B	
126: _FSQR	EQU	\$FE6C	
127: _FPI	EQU	\$FE6D	
128: _FNP1	EQU	\$FE6E	
129: _FPOWER	EQU	\$FE6F	
130: _FRND	EQU	\$FE70	

131: *	EQU	\$FE71	
132: *	EQU	\$FE72	
133: *	EQU	\$FE73	
134: *	EQU	\$FE74	
135: *	EQU	\$FE75	
136: *	EQU	\$FE76	
137: *	EQU	\$FE77	
138: *	EQU	\$FE78	
139: _FFREXP	EQU	\$FE79	
140: _FLDEXP	EQU	\$FE7A	
141: _FADDONE	EQU	\$FE7B	
142: _FSUBONE	EQU	\$FE7C	
143: _FDIVTWO	EQU	\$FE7D	
144: _FIEECNV	EQU	\$FE7E	
145: _IEEFCNV	EQU	\$FE7F	
146: *			
147: _CLMUL	EQU	\$FEE0	
148: _CLDIV	EQU	\$FEE1	
149: _CLMOD	EQU	\$FEE2	
150: _CUMUL	EQU	\$FEE3	
151: _CUDIV	EQU	\$FEE4	
152: _CUMOD	EQU	\$FEE5	
153: _CLTOD	EQU	\$FEE6	
154: _CDTOL	EQU	\$FEE7	
155: _CLTOF	EQU	\$FEE8	
156: _CFTOL	EQU	\$FEE9	
157: _CFTOD	EQU	\$FEEA	
158: _CDTOF	EQU	\$FEEB	
159: *			
160: _CDCMP	EQU	\$FEEC	
161: _CDADD	EQU	\$FEED	
162: _CDSUB	EQU	\$FEEE	
163: _CDMUL	EQU	\$FEEF	
164: _CDDIV	EQU	\$FEF0	
165: _CDMOD	EQU	\$FEF1	
166: _CFCMP	EQU	\$FEF2	
167: _CFADD	EQU	\$FEF3	
168: _CFSUB	EQU	\$FEF4	
169: _CFMUL	EQU	\$FEF5	
170: _CFDIV	EQU	\$FEF6	
171: _CFMOD	EQU	\$FEF7	
172: _CDTST	EQU	\$FEF8	
173: _CFTST	EQU	\$FEF9	
174: _CDINC	EQU	\$FEFA	
175: _CFINC	EQU	\$FEFB	
176: _CDDEC	EQU	\$FEFC	
177: _CFDEC	EQU	\$FEFD	
178: *			
179: _FEVARG	EQU	\$FEFE	>d0,d1
180: _FEVECS	EQU	\$FEFF	d0,a0>d0
181: .list			

突然ですが 創刊100号記念の予告です 激突 祝一平vs西川善司

若手実力ナンバーワンの西川善司が満開製作所の祝一平氏に挑戦状をFAX送信!

通信ケーブルに炸裂する地震、洪水、火山
そして毒の沼に落ちるのはいずれか?
編集室は闘いのワンダーランドと化す

[注意事項]

Oh!X1990年8月号は諸々の事情により定価据え置き560円(税込)となります。要するにオマケはつきません。あらかじめご了承ください。

通巻100号のなかには、本誌別冊の「ADVANCED MZ-700」(通巻35号)も含まれております。あらかじめご了承ください。

なお、次号では豪華なプレゼント企画が予定されています。あらかじめご了承ください。

たいへん申し訳ありませんが、一部お見苦しいページのある場合が考えられます。あらかじめご了承ください。

祝一平氏へ

対戦ホスピタス 決着つけましょう
5月29日 午後8時。
Xへんしゅ-ぶにて



勝てるものなら勝ってみろ
西川善司

494面のパスワードを知ってるか?
私は知ってる。

猫とコンピュータ ホットラインで

Takazawa Kyoko
高沢 恭子



盆、正月に次いで一大イベントと化したゴールデンウィーク。今年のキョウコさん一家は、パソ通仲間と電話で親睦を深めたり、近場に出かけたりなど、ゆったりと休日を満喫したようですね。

「メールなんですけどね。送ってしまっ
てから内容の修正をしたいとき、どうすれば
いいんでしょう」

「メールですか？ メールは……」

「あの、PC-VANなんか、まちがえたとき
はサブメニューが出て訂正できるようにな
ってるんですね」

「あ……、FBIはそういうコマンドはあり
ませんねえ……あらためて送りなおすしか
ないんです。私もそうしてますけど」

「そうですかあ、あと、アップロードした
自分の書き込みを消したいときなんですけ
ど。このあいだご主人にうかがったんです
が忘れてしまひましてね」

「あれはタイトルの頭にアスタリスクを
をつけて、CRすればいいそうですよ」

「いやあ、やってみたんですが、まったく
受けつけないんです」

「じつは私も前に失敗したもので、使って
ないんです。主人が帰りましたらすぐお電
話しあげます」

「いや、どうもどうも」

熟年のパソコンマニア、コバヤシカズヨ
シさんが半年ほど前から通信をはじめられ
て、今日もあいにく夫の帰宅前に質問の電
話をいただいた。パソコンのご趣味は長く、
ずっとひとりだけでいじっておられたが、1年
くらい前に日経パソコンに掲載されたクラ
ブ紹介の記事を見て「きまぐれコンピュー
タクラブ」に入会された。ここでクラブの
ボードがあるFBIにも加入されたのだ。

パソコンにくわしい人でも、通信はネッ
トごとにマニュアルがさまざまなので、わ
かりにくいこともたくさんある。そんなと
きはコバヤシさんのように、さっさとメン
バーに聞いてみるのがいちばんいい。でも、
パソコン族はこういうダイレクトな手段を
えらばない人が多いみたいだ。

帰宅した夫にコバヤシさんの質問を伝え
ると、彼はすぐにFBI-NETにアクセス。
「きまこん」のボードにひとつ書き込みを
したあと、「消去」の実演をしてダウンロード。
その実録を入れた手紙をコバヤシさん
宛にMAILした。それから電話をして、
MAILBOXを見てくださいと伝えた。

🐾 ホストが消えた

「あの……またちょうど仙台に出かけて
まして」

なぜかコバヤシさんからのお電話に、夫
はいつも不在だ。

「そうですか。いや、FBIがこのあいだの
日曜日から通じないんです」

「あ、いまシスオペのお宅はお引越しの
直後なので、システムは閉じてるんです。
CLOSEは1週間の予定だそうですけど」

「そうなんですかあ。コバヤシ先生もつな
がらないってふしぎがってましてね」

同じコバヤシさんでも、「目白のヤス」こ
とコバヤシヤスジ先生は大学の先生で、こ
ちらのカズヨシさんは元お役人だ。

「広報室のボードに『お知らせ』があった
んですけど……」

「だって、そんなとこ見ませんもの」

ちょっと不満そうにされたのが、とても
率直でかえって好感をもってしまう。

「新しいインテリジェントビルの誕生で
す。ナカムラシスオペ個人のお宅ですけど
きれいな白いビルだそうです」

「そりゃあ、花束でも持っていかなきゃい
けませんな、ハハハ」

明るい声に、私も電話のこちら側でうれ
しくなってきた。なんだかFBI-NETその
ものがバージョンアップして新装開店する
ような気分だ。

当のナカムラ隊長のお宅では、そう浮か

れてばかりはいられない大移転のようだった。
ゴールデンウィークの過ごし方は、ニ
ュース番組の主要なタイトルのひとつにも
なっていたが、隊長一家の黄金週間はなん
と引越したったわけだ。5回線ぶんの設備
の移動だけでもたいへんなのはたやすく想
像できるが、その上、生活一式を運ぶのだ。
じっさい、段ボール30箱ほどつめても、ま
だマシンが入りきらないとは、準備中の隊
長の弁だった。

🐾 23番めの区

わが家のゴールデンウィークは、「わが家
を基地にしてゆったりと過ごす」のがテー
マだった。当初の予定だった北海道旅行は
早いうちにキャンセルした。これは実行し
ていれば雨の日に当たっていたので、運の
よい判断だったような気もするけれど、出
かけた場合のお天気はまた違うのかもしれ
ない。

トオルの希望で銀座のドイツ料理を味わ
ったり、近くの葛西臨海公園に遊んだりし
た。公園内にある話題の東洋一という水族
館も、お友だちと出かけたことのあるトオル
の推薦だったが、休日の超人気で入館は
あきらめなければならなかった。

千葉県にもっとも近い東京の東端のわが
区は、新しい都市としての発展がめざまし
い。都心が人の住まうところとしては廃墟
になっていくかわりに、このあたりの人口
はどんどんふくらんでいる。それにあわせ
て、区の町づくりの意欲はたいへんなもの
だ。

小学生のころ23区を暗記するとき、番号
がついていた。1 千代田、2 中央、3 港、
4 新宿、…この区は23番めだった。でも世
田谷とならんで面積が大きい。だからいま
こうして、公園、施設、広場、鉄道もでき

る。店舗の種類も数もじゅうぶんなうえに、さらに建設中のビルが、ひとつの視野に10あまりも見える。

わが家のあるN町のとなり、S町とR町の人口増加はたいへんらしい。川沿いから海に向けて広がってこうとしている高層団地の町である。古い町を活性化させるのではなく、新しい町を創造していく人たちは、いまとても努力をしている。教育、文化、地域のコミュニティなど、町全体で高めておたがいをはぐくんできようとしている。ちなみにS町の団地内にある中学校のひとつは、都内一の学力を持っているのだそうだ。

ゴールデンウィークが明けてすぐ、この2つの町が主催した「ふれあいまつり」があった。団地内の子供会や、300以上もあるさまざまな趣味の同好会が力をあわせて、広場のプロムナードにたくさんの催しをくりひろげていた。リサイクルの店、伝承あそびのコーナー、健康相談。古本市、金魚すくい、伝統工芸の展示。手作りみこしや鼓笛隊のパレードもあり、石炭をたいたミニSLも走っている。その中で、NTTコーナーの立て看板が、「ひつじ年から都内の局番が4ケタになります」と、にぎにぎしさを象徴するように呼びかけていた。

運営委員会と思われる方たちはそろいのTシャツで、きびきびとスタッフをつとめている。どの人も、みずからが楽しんでいるようすで、これがおまつりを盛り上げるいちばんの力にちがいないと思った。きつとよく話し合いが持たれて、みんなが目標を共有しているからだろう。

空を見上げると、垂直にいくつもいくつも、マンションの棟がそそり立っている。こういったマンモス団地の場合は、となりの棟まで行くにもけっこう距離があるものだ。たぶんパソコンの同好会もあるにちがいないから、そのうち自治会活動もパソコン通信などを活用したものになっていくのだろう。

そのS町とR町は、いま人口急増のために、幼稚園、小学校がパンク状態になっているようだ。以前に新設されたものもすでに満杯で、2つの町の父母たちは増設の運動を熱心にすすめているが、子供の数は確実に減っていくことを主張する区側はこぼれている。1クラスの数が多いすぎて、お

友だちの名前がおぼえきれないほどの幼稚園もあって、数が多いことの弊害が深刻にうたえられている。

ウイルスの害

ゴールデンウィークはX68000のウイルス報道も話題になった。用語だけは取り揃えてあったけれど、なんだか筋書きが未熟すぎるので、奇妙に思った人はたくさんいたと思う。

でも、パソコンに馴染みのない人にとって

は、さぞものものしいできごと映ったことだろう。自分の知らない世界については、どんな奇抜な想像があってもおかしくない。プログラムの「虫」というのは、バクテリアのようにほんとうにマシンを食いあらすと思っていた、とジョークまじりの新聞の投稿を読んでも、とても共感できる。

私は逆に、むかし祖母や母が話す「衣類の虫」というのは、布が古びたりいたんだりすることだと思っていた。それがあるとき、ほんとうに生きた虫が衣類を食べて穴をあけてしまうのだと知って、びっくり仰天した。「本の虫」というのも、本好きの人のたとえに使われることもあって、じっさいに生きた虫も住みついていると知ったときは、まったく驚いた。

泣き虫、弱虫、カンの虫。「虫のいどころ」「虫が好かない」「虫の知らせ」なんていうものだから、衣類の虫も本の虫も抽象的なものだと思いこんでいたのだ。

コンピュータウイルスも、マシンにさわっていると感染すると信じている人がいるそうだけれど、解釈によっては、まったくのまちがいはいえないと思う。

人間があやつることでパソコンは動き、ウイルスの汚染も人間が操作することで行われる。私たちは感染の経路にあるし、マシンとの一体感も持っている。パソコン通信などの、信頼をベースにしているはずのコミュニケーションからウイルスをうつされるとき、精神の痛手もバカにできない。これは人間も感染したと考えられないこと



はない。「ウイルス」(virus)を辞書でひくと、①「濾過性病原体」、②(道徳的腐敗の)もと、害毒、となっている。

シャイが主流?

「あのですね、13H OPENというのでアクセスしてみたんですけど、FBIはまだ通じないんですよ」

コバヤシさんからのお電話に、やっと夫が出ることができた。

「ああ、少し遅れているらしいです。その間の情報は、ナツメの『千夜一夜』のボードにだいたい書かれていますから……。ええとアクセスポイントは……」

それから3日ほどして、例の明るい声。

「やっとな復活しましたねえ、前のおりなのでホッとしました。ナツメは入って見ましたが、なんだかわからないことばかり書いてあって、ハハハ」

5月16日にFBIは再開した。ポートピープルの気分だったというメンバーを、一時的に収容したNATUME-NETの「千一」ボードは、一気に100以上のアーティクルが増えた。

10日ばかりのシステム停止で、私たちもホスト局のありがたががよくわかったが、その一方、コバヤシさんからの何回かのお電話で、FBI不在のあいだもほのぼのと楽しかった。閉鎖的や自分本位の代名詞にされがちなパソコンマニアだけれど、ほんとはみんなコバヤシさんのようにやっていたのだ。でも、ちょっとガマンしている。

BACK ISSUES

バックナンバー案内

ここには1989年7月号から1990年6月号までをご紹介します。現在1989年6～12、1990年1～6月号までの在庫がございます。バックナンバーおよび定期購読のお申し込み方法については、168ページを参照してください。

1989



7月号

特集 3Dグラフィックへの飛翔
Zバッファアルゴリズム/スムーズシェイディング 他
THE SOFTOUCH Terazzo PRO-68K/アドヴァンスト・ファンタジアン
新連載 DōGA・CGアニメーション講座
MZ-2500用グラフィックエディタ作成講座
マシ語カクテル in Z80's Bar
X-BASICプログラミング調理実習
全機種共通システム TTC用パズルゲームTIC BAN
X68000マシン語プログラミング/C調言語講座PRO-68K 他



8月号

特集1 X1プログラミングガイドブック
PCGの基礎から奥義まで/超高速ラインルーチン 他
特集2 3Dグラフィックの深淵へ
スキャンラインバッファ/3Dモデリング 他
新連載 (で)のショートプロバート
X68000マシン語プログラミング/C調言語講座 PRO-68K
X-BASICプログラミング調理実習/DōGA・CGA講座
MZ-2500用グラフィックエディタ/Z80's Bar 他
全機種共通システム CP/M用ファイルコンバータ



9月号

特集 活用ハードディスク&プリンタ
各社ハードディスク接続総チェック/ハードディスク雑学
講座/COPYキーメニュー/ビデオプリンタ活用プログラム 他
THE SOFTOUCH ジェノサイド/琉球/mFORTH Compiler
●サイバースティックで遊ぶ 不思議な環境ソフトの世界
●X1/X1turbo用シューティングゲーム Defeat X
Z80's Bar/MZ-2500グラフィックエディタ 他
[X68000] X-BASIC/マシン語/C調言語講座/DōGA・CGA
全機種共通システム 生物進化シミュレーションBUGS



10月号

特集 ゲーム面白心理学
ソーサリアン・宇宙からの訪問者/ファンタジーゾーン
ねじ式/ガウディ・バルセロナの風/サバッシュ 他
●MZ-700用シューティングゲームSide Roll-F
●X1/X1turbo用カードゲームBonding
ショートプロ/Z80's Bar/MZ-2500グラフィックエディタ
X68000マシン語/X-BASIC/C調言語講座/DōGA・CGA
THE SOFTOUCH Z'sTRIPHONY DIGITAL CRAFT/James68K
全機種共通システム 小型インタプリタ言語TTI



11月号

特集 microComputer入門
初歩からのCPU物語/RISCプロセッサの設計と製作
X68000&X1で周辺LSIを使いこなそう
連載 ショートプロ/Z80's Bar/MZ-2500グラフィックエディタ
X68000マシン語/X-BASIC/C調言語講座/DōGA・CGA
●X68000用カードゲームばばぬき
LIVE in '89 メタルホーク/オブ・ラ・ディ、オブ・ラ・ダ
THE SOFTOUCH Stationery PRO-68K/リンクマスター1
全機種共通システム TTI用パズルゲームPUSH BON!



12月号

特集 Cプログラミングへの招待
付録 C言語簡易リファレンス
連載 ショートプロバート/Z80's Bar
X68000マシン語/X-BASIC/DōGA・CGA
●Oh! X2周年特別企画「素粒子の音が聞こえる」
●X1/turbo用アクションゲームACTIVE UNIT
LIVE in '89 天空の城ラピュタ/ギャラクシーフォース
THE SOFTOUCH 38万キロの虚空/た〜みのる2
全機種共通システム SLANG用ダイレクションライブラリ

1990



1月号

特集1 オペレーティングスタイルの研究
特集2 Cプログラミング応用編
連載 ショートプロバート/Z80's Bar
X68000マシン語/C調言語講座/DōGA・CGA
●X1/turbo用シミュレーションゲームSuper Battle
LIVE in '90 さよならを過ぎて/RIDEEN
THE SOFTOUCH レナム/メタルサイト
全機種共通システム WORM KUN/再掲載SLANG
特別付録 X68000 THE SOFTWARE CATALOGUE



2月号

特集 画像圧縮へのアプローチ
連載 ショートプロバート/Z80's Bar/DōGA・CGA
X68000マシン語/C調言語講座/X-BASIC調理実習
●X68000用ゲームプログラムGon Gon
●MZ-700用紙芝居Eylarath
LIVE in '90 オーダイン/魔女の宅急便
THE SOFTOUCH A-JAX/フラッピー2/夢幻戦士ヴァリスⅡ
マジックパレット/Mu-1/CYBERNOTE PRO-68K
全機種共通システム 超小型コンパイラTTC++



3月号

特集 MUSICアドベンチャー
X68000用MIDIドライバ&音源エディタ
なんでも鳴らせるOPMD.X/MMLを楽譜データに
連載 ショートプロバート/Z80's Bar/DōGA・CGA
C調言語講座/X-BASIC調理実習
●X1/turbo用シミュレーションCRISIS in Tokyo
LIVE in '90 パワードリフト/スキーム/となりのトロ
THE SOFTOUCH ナイトアームズ/斬/ダンジョンマスター
全機種共通システム 超多機能アセンブラOHM-Z80



4月号

特集 ゲームシステム文学誌
1989年度GAME OF THE YEAR発表
連載 ショートプロバート/Z80's Bar/DōGA・CGA
X-BASIC調理実習/C調言語講座/X68000マシン語
●X1・MZ-2000/2500用RPG The Cave of Dalk
●うわさの68040、ついに登場
LIVE in '90 バーニングフォース(OPMD対応)
THE SOFTOUCH The Fille Professor/HOST PRO-68K
全機種共通システム ファジコンコンピュータシミュレータ-MY



5月号

特集 BASICプログラミング
第5回 言わせてくれなくちゃだわ
連載 ショートプロバート/Z80's Bar
X-BASIC調理実習/X68000マシン語プログラミング
●新機種X68000SUPER-HD/EXPERTⅡ/PROⅡ
●ラジコンスティックの製作
LIVE in '90 TURBO OUTRUN
THE SOFTOUCH 天下統一/ボビュラス/Hyperword
全機種共通システム インタプリタ言語STACK



6月号

特集 創刊8周年記念PRO-68K(付録5"2HD)
Oh! Xアンケート結果大分析大会
連載 ショートプロバート/Z80's Bar/PurePASCAL
X-BASIC調理実習/X68000マシン語プログラミング
●X1/turbo用コマンドシェルシミュレータ
●ハードウェア入門
LIVE in '90 ナイトアームズ/悪魔城伝説/この木なんの木
THE SOFTOUCH 三国志Ⅱ/FAR SIDE MOON/グラナダ
全機種共通システム X68000用S-OS"SWORD"他



ノーマルX1への対応

亀田 雅彦 Kameda Masahiko

お待たせしました。MS-DOS風ファイルユーティリティINTEGRAL X1のノーマルX1対応ルーチンの発表です。同時にマシン語部分のソースリストも公開します。今回のプログラムを使用するには6月号のプログラムが必要ですので注意してください。

今月は、ソースリストの公開と待望のノーマルX1への対応です。これで、X1turboでなくても便利なディスク関係の命令が使えるようになりました。なお、6月号の「INTEGRAL.X1」は打ち込み終わったでしょうか？ 6月号のプログラムがいちばんの基本になるので、それがないと今月(来月)のプログラムは動きません。というか、なにがなんだかわけがわからないことでしょう。悪しからず。

それから、先月号書き忘れてしまったことを書きます。BASICプログラムの「COMMAND.X1」を打ち込むときは、くれぐれも変数名に気をつけてください。1文字違っただけでも、おNEWな変数になってしまいますから。特に、最後のほうのまとめで16進数を代入しているところは、間違えると即暴走につながりますよ(テストするとき、壊されちゃ困るディスクはドライブから抜いておくこと)。どうしても動かないじゃん! とか、プログラムのバグなんじゃないの? と思う人は、どこかでX1turboZを使わせてもらって、先月の解説を見て、付属ディスクからプログラムを作り出してください。

また、「INTEGRAL.X1」というのは6月号に掲載された基本プログラムの愛称ですし、KAME-DOSというのは、それらをベースにこれから構築されていくシステムの総称ですのでお見知りおきを。

序論

そもそもこのプログラムは、次の3つのコンセプトに基づいて設計・開発されています。

- 1) X1において、Human68kやMS-DOSとのファイル&操作性における互換性の確保
- 2) 初代X1からX1turboZにいたるまで、その能力を最大限引き出すような汎用プログラミング

3) 誰にでも拡張&改造ができるように、BASICでコマンドを作成するフレキシブルな構造

それぞれの特徴は、先月号を読めばわかります。1は、コンバータなどに代表される文字どおりの互換機能¹⁾。今月は、次の2にあるように、ノーマルX1への対応です(ちなみに来月は3をサポートする予定)。もともとX1全シリーズを念頭においていたので、変更したり新たに打ち直したりするところはわずかで済みました。

また、ハードの都合上サポートできないもの以外は、ほぼX1turbo版と同じ機能を持っています。

それでは、本論に入る前の予備知識として、KAME-DOS動作時のメモリマップを図1に示しておきます。

まず、BASICがメインメモリの半分以上を占拠していて、&HD000からのいちばんおいしい部分にマシン語領域を確保しています。そのあいだには、COMMAND.X1などの制御用BASICプログラムが置かれます。

さて、6月号でも説明したとおり、G-RAMやバンクメモリはRAMディスクとして、またはバッファ領域として確保されます。使うBASIC(や機種)によって多少メモリ構成が違いますが、INTEGRAL.X1自体は不動の4番なのでほとんど変更はありません。

そして、実際にFDCをアクセスしたり、RAMディスクをアクセスしたりするのがマシン語の役割です。読み込んだデータを処理したり、ディレクトリを検索したりするのもマシン語で行います。

BASICプログラムは、ユーザーからの入力処理(実際ただのINPUT命令だったりする)と命令ごとの振り分け、それにその命令のもっともおおまかな処理を引き受けます。もちろん、BASICプログラムやマシン語だけではなんの役にも立ちません。それぞれが密接に連絡しあってひとつの命令をこなしていきます。

今月解説するのは、このマシン語の部分です。外部コマンドを発表して、ユーザー自身がそれを拡張していくには、どうしてもBASICからマシン語を呼び出していくことになります。そのためにも、ソースリストをよく見ておいてください。なお、使用したアセンブラはS-OSのREDAです。

そして、もうひとつはX1用のルーチンです。X1turboの方は、このルーチンは必要ありません(CZ-8FB01上で動かすときも必要ない)。機種別・BASIC別になにが必要でなにがいらないか? などは、あとでまとめて書いておくので参考にしてください。

図1 メモリマップ

0000	BASIC システム領域
	BASIC プログラム領域
D000	ジャンプテーブル1
D060	ワークエリア1
D1A0	プログラム領域1
DF00	階層ディレクトリ用 ワークエリア
E000	ジャンプテーブル2
E051	ワークエリア2
E100	固定データ
E200	プログラム領域2
EC40	デバイスドライバ 登録テーブル
EC80	バス指定
ECC0	変数保存エリア
ED00	フリーエリア
EE00	汎用バッファ
EF00	フリーエリア
F000	ノーマルX1用 ディスクアクセス関係 turboではシステム

ノーマルX1用ルーチン

まず初めにX1用の打ち込み方と、使い方を説明しましょう（くどいようですが、X1turboでは以下の操作は必要ありません）。リスト1がダンプリストです。ソースリストはX1turbo用と一緒にになっています。BASIC(CLEAR&HD000を実行すること)やS-OSから打ち込んでください。終わったら、6月号のマシン語ルーチン(&HD000~&HEDFF)をロードして、

SAVEM "FDC.OBJ", &HD000, &HF151
として、一緒にセーブしておいてください。今後ロードするときは、こちらのほうをロードします。

そうしたら、このルーチンを登録しなければなりません。これまた、6月号の「INTEGRAL.X」というコンフィギュレーションファイルを大幅に変更します。新しく作り直したほうが早いので、後ろにまとめて説明しているところにリストを載せておきます。これを打ち込んで、6月号のINTEGRAL.Xに差し換えてください。

ここまでくれば、登録作業は完了です。あとはX1turbo用と同じように使用できます。違いがあれば、その都度フォローしていきます。

現時点でX1でできないことは、2HD、バンクメモリ、F:(MEM1:)デバイス、漢字表示です。階層化ディレクトリはKAME-DOSではサポートしていますが、BASIC側がサポートしていないので注意してください。

それから、「COMMAND.X1」の中の注

釈は打ち込まないようにしてください（行は残してコメントを入れない）。X1turboでも、CZ-8FB01を使うときは同じです。実行中、メモリ容量不足で止まることがあります。

データ処理論

では、解説を始めましょう。なお、この連載の中ではプログラムの解説よりも、「いかなるテクを使っているのか」というアルゴリズム関係を中心に、なるべく図と言葉で説明していこうと思います（解析したい方は、ソースを見て）。ですから、他機種のユーザーの方も「X1の記事は飛ばそう！」などといわずに、読み物として「こういう手法もあるんだなあ」ぐらいに思っておくと、将来役に立つかもしれません。

ハードウェア攻略法

DOSとは切っても切れない関係にある、ドライブまわりから攻めてみましょう。ディスクドライブの制御には、必ずFDC(フロッピーディスクコントローラ)という石が、CPUとドライブの仲立ちに入ってきます。こいつに命令を与えることで、間接的にモーターやヘッドを動かしたり、CPUとのあいだでデータの受け渡しを行ったりします。

そして、もうひとつ重要なのがDMAC(ダイレクトメモリアクセスコントローラ)という石です。詳しいことは参考文献にゆずるとして、X1turbo²⁾ではこれがCPUの代わりに、FDCとデータのやりとりをして

います。なぜ、こんな石を使うのか？ というと、ソフトウェア側でタイミングを取る必要がなくなるからです。たとえば「256バイト読め」という命令をあらかじめ与えておいて、「行け!」と指示を出せば、ドライブ・FDC・DMACのあいだで勝手にデータを読み込んでおいてくれるので、非常に便利です。

その他のデバイス(EMMやMEM)に対する入出力は、すべてCPUがシャコシャコと1バイトずつ転送します。また、画面やキーボードなどのコントロールは、当然のごとくBASICまかせです。

ディスクに関する用語、セクタ・トラック・レコードなどは、囲みて簡単に説明しておきます。詳しく知りたい方は、ユーザーマニュアルや参考文献を参照してください。

ソフトウェア攻略法

KAME-DOSのシステムは、MS-DOSやCP/Mと同じように、機能別に3つの部分に分かれています。

- 1) 最下層に位置し、ハードウェアに密着してバイト単位で記憶装置からのデータの入出力を担当する部分。レコード単位の処理が多い(リスト3)
- 2) 主に1を利用した、入出力データの管理。あるいは、コマンド実行ルーチンの集合体。3から呼び出されることが多い。1の段階では、どのデータも等価だが、ここでディレクトリ・FAT・ファイルなどの意味を付加する(リスト2)
- 3) コマンド解釈・実行部分。マンマシンインタフェイスでもある。KAME-DOSではBASIC(6月号の「COMMAND.X1」)

それぞれの対応を図2に示しておきましょう。そして、上記の3つのほかに、「デバイスドライバ」というルーチンもあります。これは、ユーザーがあるデバイスを制御するとき、そのデバイスに対して入出力を行うプログラムのことです。通常、起動時

ディスク用語

トラック：5インチフロッピーディスク上には、バームクーヘンのように数十本の「輪」があって、それをトラックと呼ぶ。2Dであれば、表裏にそれぞれ40本ずつ、計80本あるのが普通である。

セクタ：各機種でフォーマットするときには、そのトラックをまたいくつかの部分に区切る。区切ったひとつ分をセクタと呼ぶ。実際のデータはこのセクタ上に記録されるので、セクタ長とセクタ数がディスク容量を決定する。また、ディスクは1バイトずつデータをアクセスできるわけじゃなくて、このセクタごとには読み書きできない。1バイト入出力はソフトの仕事である。

レコード：デバイス上で読み書きできる最小単位のこと。すなわち、フロッピーディスクではセクタとなる。

クラスタ：レコード単位で読み書きできるけれど、それじゃ面倒くさいときにエイヤツと何レコードかをまとめて読み書きするための単位。

ファイルは、このクラスタがいくつかつながったものとして管理されている。

ディレクトリ：ファイル名などのファイルごとの情報が詰め込まれているテーブル。

FAT：ディスク上のクラスタ間のつながり具合を管理しておくテーブル。これが破壊されると、ディスクアクセスができなくなる重要な部分。ディレクトリとともに、ディスク上の決まった場所に書き込まれている。

MS-DOSのフォーマットを紹介しておく。

	2HD	2D
片面トラック数	77	40
セクタサイズ (単位はバイト)	1024	512
1トラック当たりの セクタ数	8	9
1クラスタ当たりの セクタ数	1	2

参考：「MS-DOS3.3プログラミング&リファレンス」
平沢正之 日本ソフトバンク

注1) 実際に私も、日本語のコメントをソースリストにつけるために、S-OSファイルをX68000上に移してEDで編集した。もし読者が適当なMS-DOSディスクを持っていない人は、付録ディスクでDIRやTYPEを試してみよう。

注2) X68000やPC-9801でも、同様の理由でディスクアクセスにはDMAが使われている。アクセス中に音楽が鳴ったりできるのも、CPUがそれだけ楽をしているから。

ノーマルX1では余計なハードは一切ついてないので、CPUがデータの読み書きまで担当している。

にDOSを組み込んだり、切り離したりします。すなわち、DOSは初めから「ディスクをドライブするデバイスドライバ」を組み込んであるので、ディスクアクセスができることになります。

KAME-DOSでは、ディスク(含むRAMディスク)以外のドライバは基本的に BASICでの処理になります。もちろん設定を変更してマシン語にすることはできますし、登録を増やすこともできますが、それではKAME-DOSの意味がなくなってしまいます。機会をみてデバイスドライバの拡張も行ってみましょう。囲みにデバイスドライバの登録方法を書いておきます。

さて、外部からマシン語を呼び出す方法ですが、ルーチンごとに必要なワークエリアに値をセットしてコールする方法です。一般的なDOSコールのように、パラメータをレジスタ渡しにする方法ではありません。BASICでの制御なので、このようにしました。コールアドレスおよびワークアドレスはまとめて設定してあるので将来のバージョンアップにも対応可能です。

また、エラーが発生したときには、処理を中断してエラー番号をワークに設定し、BASICの制御下まで戻ってきます。したがって、どこでエラーが起きたとしても必ず「COMMAND.X1」内のエラールーチンにくるようになっていきます(これを利用すれば、標準エラー出力なんて処理もできるでしょう)。もちろん、BASICのON ERROR GOTOなんぞは一切使用しておりません。

バッファ攻略法

6月号でもちょっと触れておきましたが、ディスクをアクセスする場合には必ずある程度のバッファをメモリ上に取る必要があります。あるデバイスからのデータは、まずこのバッファに格納されて、そのあとで指示されたデバイスへと送り込まれます。たとえば、

COPY A:ABC B:
とした場合、まずバッファいっぱいまで、Aドライブから「ABC」というファイルを

図2 プログラム相関図

KAME-DOS	LIST2	LIST3
COMMAND.X1		
CP/M	BDOS	BIOS
CCP		
MS-DOS	MS-DOS.SYS	IO.SYS
COMMAND.COM		
Human68k	Human.SYS	Human.SYS
COMMAND.X		

読み込みます。そして今度はそのデータを、バッファからBドライブへ書き込みます。「ABC」というファイルの容量がバッファより小さければ、1回の「読み書き」で終わりますし、大きければその分だけアクセス回数も増えてしまうでしょう。

「なぜ、こんな面倒なことをするのか? 読み書きするメモリ間で直接データをやりとりすればいいじゃん」と、思う人もいるかもしれません。でもそれじゃ、アクセスするデバイスの数だけプログラムが必要になって、能率的ではありません。入力も出力も「すべての道はバッファに通ず」としておけば、どんなデバイスであろうと、そのバッファに対してアクセスするプログラムがあればいいのです。便利。便利。

実は、デバイスドライバという手法も、この考え方からきています。これも図3に示しておきましょう。

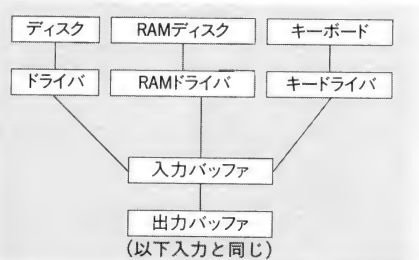
ちなみに、X1シリーズのBASICでは、バッファは256バイト固定です。これはとりまなおさず、X1フォーマットの1レコード(1セクタ)に当たります。MS-DOSやHumanでは、CONFIG.SYSのなかで、

BUFF=20 1024
などとして、設定できるようになっています。1KバイトというのはMS-DOSフォーマットの1レコード(1セクタ)の長さです。つまりバッファの大きさは、1レコードの倍数になっているのが普通なんです。そして、X1の256バイトというのはとんでもなく小さいので、ディスクアクセスも遅いのです。

KAME-DOSでは最低バッファ容量が4Kバイトになります。これは、X1の1クラスタ、MS-DOSの4レコード分に当たります。これだけのデータをまとめて読み書きするので、理論上BASICの16倍のスピードということになりますか。でも、実際にはRAMディスクアクセス時はバッファが小さいほうが速いので、どちらがいいともいえません。

このように、DOSというのはデータの仲介業者だったりします。自分でも、ディスクをアクセスするルーチンを持っています

図3 バッファ&デバイスドライバ



が、そういうデバイスドライバ同士のあいだに立ってのデータの受け渡しが本業といえるでしょう。バッファはいわば倉庫であり、デバイスドライバが荷物を運ぶトラックやコンテナで、DOSはそれを仕切っている会社です。それじゃ、人間はなんでしょう? その会社の社長といえるくらい、DOSを使いこなしていますか?

プログラム分析論

DOSにはいろいろな命令がありますが、マクロ的に見ればそれらの処理がだいたい似ていることがわかります。人間にとっては、BASICファイルとワープロの文書ファイルは全然意味が違いますが、DOSにとってはどちらもただのファイルです。データ(ファイル)の制御という観点から、KAME-DOSの動作概念と個々のルーチンワークを調べてみましょう。

●動作概念

DOSの本質的な動作原理は、入力・処理・出力の3つに大別されます。つまり、ほとんどの命令はこの一連の動作の繰り返しなのです。たとえば、

TYPE A:ABC
という命令は、Aドライブから「ABC」ファイルを(バッファに)入力して、「ファイルの終わりかどうか? の判断」という処理をして、(バッファから)画面に出力します。入出力はそれぞれのデバイスドライバが担当しますが、「処理」の部分はDOSが行います。エンドコードの検出のほかにも、X1・MS-DOS間の改行コード変換や、ディスク上のディレクトリからディスプレイ形式への変換作業などがあります。

その他、いくつかの注意点ががあります。入出力するときには、BASICと同じようにファイルOPEN・CLOSEを行います。バッファには入力用(I)と出力用(O)の2つがあって、上記のようなデータ変換を行う場合は、

デバイスドライバの登録法

&HEC40から、ひとつのデバイスごとに6バイト割り振られていて、全部で8個登録できます。そして、そのうち5個は設定済みなので残りは3個です。6バイトの内訳は、最初の3バイトがデバイス名(NUL, CONなど)、次の1バイトがBASICかマシン語かの設定(0がBASIC, 1がマシン語)、最後の2バイトはマシン語のジャンプアドレスです(BASICのときは無関係)。4, 5, 6番目のデバイス名は、見てわかるとおり「\$1@」などと架空の名前になっているので、これを書き換えられます。なお、この架空の名前と同じファイル名は指定できません。

「デバイスからデータをIに入力」

「Iから1バイトずつ読んで、変換して、Oに書き込む」

「Oからデータをデバイスへ出力」という手順を踏むので多少遅くなります。もちろん、変換をしないCOPYなどはIから直接出力するので速くなります。

MS-DOSとX1フォーマットの取り扱いについては、その構造の類似性を利用して、違うところだけをデータ化して対応しています。FATに関しては個別のプログラムで処理しています。またセクタ長の違いも、バッファに入れるときにつじつまを合わせておいて、あとは入力したバイト数でのみ操作する構造にし、ほかのルーチンに余計な負担がかからないようになっています。

それでは、ここから面白そうな個別ルーチンを拾って見てみます。まずはリスト3(ディスクアクセスの基本ルーチン集だよ)から。

●RWREC

1レコード分のデータ入出力。

ディスクをアクセスする際には、必ず呼び出される使用頻度のきわめて高いルーチン。RAMディスクとフロッピーディスクではまったく違うルーチンへ分岐するので、2つに分けて説明します。

A: フロッピーディスク

DMAを使っているので読むのと書くのではそれほど差がなくて、DMA用データとコマンドを変更しているくらい。フォーマットによって違うのはセクタ長だけ。

基本動作として、「バッファのバンクをアクティブにする・ターゲットセクタを算出する・モーターを回し、トラックをシークする・FDCにコマンドを与える・DMAを走らせる・(DMAが勝手に入出力)・モーター停止・バンクをもとへ戻す」の一連の動作をして、そのあいだにエラー判定が入ります。面白いのは、1レコード処理時間内(100分の1秒単位)にこれだけやってしまうことです。もし遅れれば、次のセクタが(ヘッドの下を)通りすぎていて読めなくなり、最高で26倍(X1の2HD)しかも次のセクタでもそれが起こるので、1クラスタ(X1で16セクタ)で最高26×16倍遅くなるのでした。

BASICのディスクアクセスが遅いのは頻繁にこの現象が起こるからで、狭いバッファの弊害がここにも出ています。

B: RAMディスク

基本的には「デバイス&バッファバンクの設定・LD命令のループ」の2つだけです。LD命令は、バッファやデバイスによってIN

/OUT命令に自己書き換えされます。そして、面倒なのがデバイスとバッファのバンク関係の把握です。表裏バンクの関係であれば、1バイト転送するごとにバンク切り換えをしますし、そうでなければただの読み書きです。たとえば、G-RAM0(E:)

にバッファを設定して、G-RAM1(F:)をアクセスすると、遅くなるのはそのためです(これを避けるにはバッファのバッファ!!? が必要)。

●CRSRW

ご存じ! クラスタ読み書き。

入力方法

X1シリーズ各種でKAME-DOSを使うためには、CONFIGファイル(ファイル名は「INTEGRAL.X1」)を書き換える必要があります。ノーマルX1、ノーマルturbo、バンク増設をしたturbo、ノーマルZ、バンク増設(内蔵)のZに分けて、それぞれに標準的なCONFIGファイルを設定しておきます。しかし、「turboでCZ-8FB01を使いたい」などというふうには、ここで紹介していないことをしたいときは、6月号を見て自分で設定してみてください。

ノーマルX1: LIST1 (使用BASICはCZ-8FB01 ver 1.0)

プログラムの内容については、6月号を見てください。1250行の、
POKE &HD07F, 0
というのは、「BASICで階層ディレクトリが使える

ない」という意味のフラグです。

バンク増設をしていないturbo: LIST2 (使用BASICはCZ-8FB02)

バンク増設をしたturbo: LIST2+LIST2.1 (使用BASICはCZ-8FB02)

LIST2を、LIST2.1の変更点に従って変更してください。

バンクを増設(内蔵)していないZ: LIST2+LIST2.2 (使用BASICはCZ-8FB02)

LIST2を、LIST2.2に従って変更してください。ちなみに、6月号に掲載したのはこの設定のものです。

バンクを増設(内蔵)しているZ: LIST2+LIST2.1+LIST2.2 (使用BASICはCZ-8FB02)

LIST2を、LIST2.1およびLIST2.2に従って変更してください。

リスト1

```
1000 '
1010 'INTEGRAL X CONFIG FILE LIST 1
1020 ' (C) M.Kameda 1990
1030 '
1040 WIDTH 80:OPTIONS=SCREEN 2
1050 DEFINT a-z:CLS 4:INIT:SCREEN
1060 PRINT "KAME DOS [INTEGRAL.X1] ver 1.0"
1070 PRINT " Copyright (C) 1990 M.Kameda"
1080 CLEAR &HD000:LOADM "FDC.OBJ":LOADM "DOS3.OBJ"
1090 m_rwrec=&HE000:MEM$(m_rwrec+1,2)=MKI$(&HF000)
1100 m_wttro=&HE01E:MEM$(m_wttro+1,2)=MKI$(&HF003)
1110 v_rscmd=&HE0A6:v_skcmd1=&HE0A7:v_skcmd2=&HE0A8
1120 POKE v_rscmd,2:POKE v_skcmd1,&H1A:POKE v_skcmd2,&H1E
1130 'POKE v_rscmd,0:POKE v_skcmd1,&H18:POKE v_skcmd2,&H1C
1140 v_smacs=&HE0AB:POKE v_smacs,3
1150 v_wait=&HE0AC:POKE v_wait,&H10
1160 v_ctrl=&HD18E:s_path=&HEC80:s_s=&HECC0
1170 s_mac4=s_s :s_tr4=s_s+26 :s_dn=s_s+30 :s_buff=s_s+31:s_ff=s_s+33
1180 s_bsiz=s_s+35:s_escp=s_s+37:s_iomm=s_s+38:s_eof=s_s+39 :s_eof3=s_s+40
1190 FOR i=s_mac4 TO s_mac4+26:POKE i,3:NEXT:POKE s_tr4,0,0,0,0
1200 INIT "MEM0:":INIT "MEM1:"
1210 MEM$(s_ff,2)=MKI$(&HD000):MEM$(s_buff,2)=MKI$(&HE000)
1220 MEM$(s_bsiz,2)=MKI$(&H1000):POKE s_iomm,1
1230 POKE s_escp,&H12:POKE s_eof,26:POKE s_eof3,26:POKE v_ctrl,32
1240 ps="e:/w:/a:/b:/;"+CHR$(0):MEM$(s_path,LEN(ps))=ps
1250 POKE s_dn,0:POKE &HD07F,0
1260 RUN "COMMAND.X1"
```

リスト2

```
1000 '
1010 'INTEGRAL X CONFIG FILE LIST 2
1020 ' (C) M.Kameda 1990
1030 '
1040 WIDTH 80,25,0,2:KLIST 1:OPTIONS=SCREEN 4
1050 DEFINT a-z:CLS 4:INIT:SCREEN
1060 PRINT "KAME DOS [INTEGRAL.X1] ver 1.0"
1070 PRINT " Copyright (C) 1990 M.Kameda"
1080 CLEAR &HD000:LOADM "FDC.OBJ"
1090 m_rwrec=&HE000:MEM$(m_rwrec+1,2)=MKI$(&HF000)
1100 m_wttro=&HE01E:MEM$(m_wttro+1,2)=MKI$(&HF003)
1110 v_rscmd=&HE0A6:v_skcmd1=&HE0A7:v_skcmd2=&HE0A8
1120 POKE v_rscmd,2:POKE v_skcmd1,&H1A:POKE v_skcmd2,&H1E
1130 'POKE v_rscmd,0:POKE v_skcmd1,&H18:POKE v_skcmd2,&H1C
1140 v_smacs=&HE0AB:POKE v_smacs,3
1150 v_wait=&HE0AC:POKE v_wait,&H10
1160 v_ctrl=&HD18E:s_path=&HEC80:s_s=&HECC0
1170 s_mac4=s_s :s_tr4=s_s+26 :s_dn=s_s+30 :s_buff=s_s+31:s_ff=s_s+33
1180 s_bsiz=s_s+35:s_escp=s_s+37:s_iomm=s_s+38:s_eof=s_s+39 :s_eof3=s_s+40
1190 FOR i=s_mac4 TO s_mac4+26:POKE i,3:NEXT:POKE s_tr4,0,0,0,0
1200 INIT "MEM0:":INIT "MEM1:"
1210 MEM$(s_ff,2)=MKI$(&HD000):MEM$(s_buff,2)=MKI$(&HE000)
1220 MEM$(s_bsiz,2)=MKI$(&H1000):POKE s_iomm,2
1230 POKE s_escp,&H12:POKE s_eof,26:POKE s_eof3,26:POKE v_ctrl,32
1240 ps="w:/a:/b:/;"+CHR$(0):MEM$(s_path,LEN(ps))=ps
1250 POKE s_dn,0:POKE &HD07F,1
1260 RUN "COMMAND.X1"
```

リスト2-1

```
1210 MEM$(s_ff,2)=MKI$(&H0):MEM$(s_buff,2)=MKI$(&H1000)
1220 MEM$(s_bsiz,2)=MKI$(&H3000):POKE s_iomm,3
```

リスト2-2

```
1120 'POKE v_rscmd,2:POKE v_skcmd1,&H1A:POKE v_skcmd2,&H1E
1130 POKE v_rscmd,0:POKE v_skcmd1,&H18:POKE v_skcmd2,&H1C
1140 v_smacs=&HE0AB:POKE v_smacs,1
```


あらかじめ読み込んでおいたFATを参照しながら、バッファ一杯までデータを読み書きします。レコード長と1クラスタ内のレコード数はフォーマットによって違うので、前もって指定してあるカウンタを使います。RWREC・「クラスタからレコードを計算する」・「FATのつながりを追う（空きFATを追う）」という重要なルーチンを使っているの、要CHECK！ また、1クラスタごとにFATを追っかけるアルゴリズムなので、えんえんと書き込みをした挙句にDISK FULLエラーになる可能性もあります。本当なら実行する前に、きっちりとFATを計算しておくべきなのですが……。

ところで、このルーチンの頭のところでトラックをSEEKしている意味不明の部分があります。いろいろテストしてみると、「2HDモードで連続的に1トラックだけSEEK」するとシークエラーが発生するようです。そのため、ここで2トラック分SEEKして（あとですぐ戻すけど）ごまかしていますが、どうもこれは釈然としませんね。

●DEVICE

FATを読み込みつつ、そのディスクのフォーマットを判定しております。読み込み中にエラーが発生したり、FATの内容が正でなかったりすると、違うフォーマットと判断されます。また、ここで読み込まれたFATが、あとにCRSRWで使われます。

データの扱い

以上でリスト3については終わりにして、リスト2に移ります。ここではハードウェアの接待ではなくて、データを中心とした処理をしています。

●DIR

その名のとおり、ディレクトリ領域を読み込んで、指定されたファイル名を探し出

します。ワイルドカードの取り扱い、サブディレクトリ、ファイル属性などというドロドロした部分があるので、解析はやめたほうがいいでしょう。フォーマット別の対応では、データ化の手法がもっとも効率的に運用されています。「DIR2」は、ほかにも該当するファイル名があるかどうか調べるルーチンです。

●PREOPEN&OPENS

BASICでいうところのOPEN。

デバイスやサブディレクトリ指定の部分の処理と、純粹にファイル名だけの処理にわけています。実際に「ファイルをOPENする」というのは、OPEN番号ごとにワークエリアを設定して、そこにファイル情報をセットすることです。ファイル情報とは、FATアドレス・バッファアドレス・ディレクトリ内での位置・フォーマット・クラスタ・サブディレクトリのクラスタなどと、ファイル名・ファイルサイズなどです。また、ファイル番号はここではコマンド列の順に0, 1, 2……です。

このルーチンで苦労したのは、ファイル名のセットのところでした。人間の不当ファイル名指定にも果敢に対応していく、というのは非常に大変です。サブディレクトリ名の判定など、あとで見てもわからないくらい汚いところです。それから、大文字/小文字の相互判定方法もCHECKポイントでしょう。ここではファイル名の代わりにデバイス名（PRN, NUL など）の判定もしています。

また、OPEN時には、「DIR」も呼び出されてディレクトリの読み込みとファイル名の検索も行っています。この時点で指定ファイル名が存在しない場合は、エラーが発生します。そして、そのディレクトリ情報をセットしたあとは、読み込んだディレクトリはそのまま捨ててしまいます。

●CLOSE

BASICのCLOSEのことです。

実際にはOPENでやったことの逆で、ワークエリアに設定しておいたファイル情報を利用して、ファイルをディスクに登録します。つまり、ディスクアクセスしてる最中はデータをクラスタごとに読み込み（書き込み）しているだけで、メモリ上のFATを使っているだけなのです。そしてひとつのファイルの処理が終わったら、もう一度ディレクトリを読み込んで、ファイル名を捜します。ここではまだクラスタに書き込んだデータを、ファイルとして登録してません（登録するということは、ディレクトリ

図6

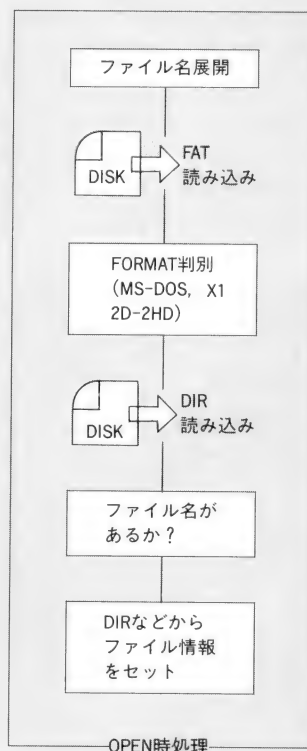
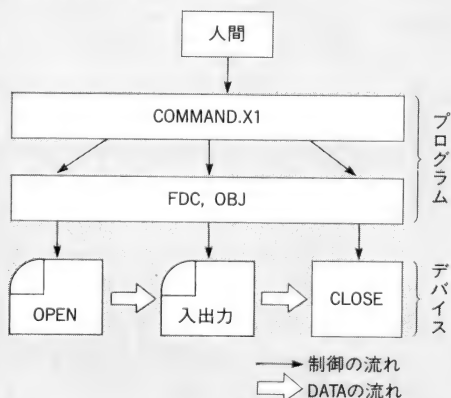


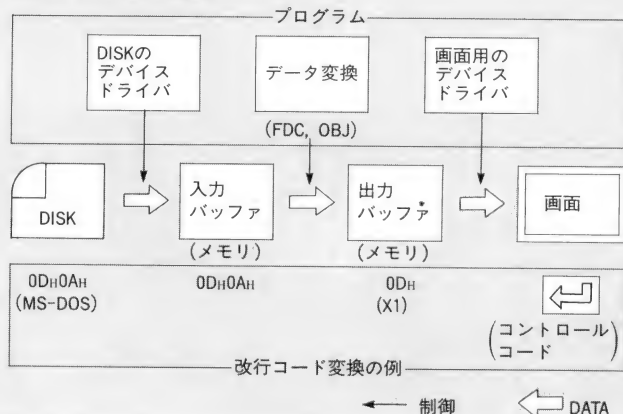
図5

図4

プログラム体系図



入出力 (TYPE) 時処理



リとFATをディスクに書き戻すことです)。

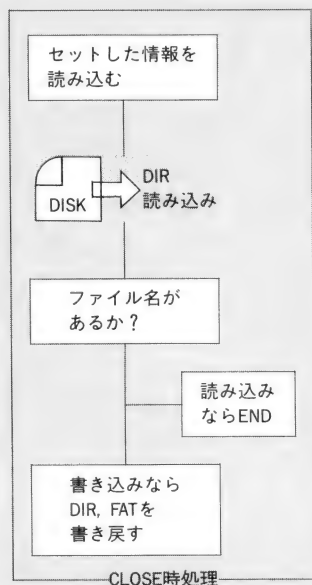
また、上書きをする場合は、ファイル名を探したときにその旧ディレクトリが見られるはずなので、そのディレクトリとFATを消してその上から書き込みます。この方法だと、新しいFATを書き込んだあとに古いFATを消去するので、FATにはぼろぼろと「穴」が空いていきますが、まあ、そういう構造なので問題はありません。

それでこのCLOSEのいちばん最後で、上記のような登録をして、やっとひとつのファイル処理が終わるのでした。

●PATH

機能は6月号で説明したとおりです。ファイルをOPENする過程で、ファイル名の前に無理やり、パスに指定されていたパス名をつけて処理します。こうすれば、あたかも人間がパスをつけて指定したのと同じことになりますね。そしてそのパス名をつけ足す作業は、ファイル名が見つかるかパス名がなくなるまで続けられます。なお、パスが有効になるのは外部コマンドを使うときなので、使い方などに関しては来月詳

図7



リスト2

```
0000 1: //////////////////////////////////////////////////
0000 2: INTEGRAL X No.1
0000 3: (KAME-DOS) for REDA
0000 4: COPYRIGHT 1990 M.KAMEDA
0000 5: //////////////////////////////////////////////////
0000 6:
0000 7:
0000 8: //////////////////////////////////////////////////
0000 9: JUMP TABLE
0000 10: //////////////////////////////////////////////////
0000 11:
0000 12: ORG $D000
0000 13:
0000 14: // JUMP TABLE /
0000 15:
0000 16: #DIR JP DIR
0003 C3 A5 D7 17 #DIRWT JP DIRWT
0006 C3 2F D8 18 #DIR2 JP DIR2
0009 C3 43 DA 19 #MSK1 JP MSK1
000C C3 7D D5 20 #ICPO JP ICPO
000F C3 6D DD 21 #TRS JP TRS256
0012 C3 0C DA 22 #DLDIR JP DLDIR
0015 C3 D3 D6 23 #CLOSE JP CLOSE
0018 C3 3C D7 24 #CLOSE2 JP CLOSE2
```

しく説明します。

●MSX1

データ変換を司るルーチン。

前に説明したとおり、入力バッファのデータを出力バッファへ転送するときに、なんらかの変換を加えます。「NOCHECK」は、なんの変換も加えないときのルーチンです。変換にも、エンドコード判定のみ・改行(コントロール)コード変換・その逆の3種類があります。どれにしてもアルゴリズムは簡単で、「1バイト読んで、必要ならひとつ前のデータも参照しながら変換して(あるいはしないで)、転送」の繰り返しです。したがって、これを任意のデータの変換ルーチンとして使うことはできません。

そして、このルーチンのもうひとつ重要な機能は、呼び出したルーチンに対して出力/入力要求をすることです。呼び出したルーチンは、この要求によって入出力バッファのデータを入出力します。また、データの終わりを知らせてくることもあります。いわば、プログラム同士で通信制御を行っているようなものです。変換の際にデータの増減が起こるので、このようなアルゴリズムになりました。

●ノーマル仕様のすごい奴

X1用ルーチンのことです。X1turboでは、

リスト1

```
F000 C3 06 F0 C3 02 F1 3A 8B : 34
F008 E0 FE 04 D2 C5 E8 C5 D5 : FB
F010 CD 78 E8 38 73 3A 81 EB : 7E
F018 01 FA 0F ED 79 16 FB 1E : 9F
F020 F8 D9 ED 4B 91 E0 D9 3A : 8D
F028 A1 E0 B7 C2 AD F0 F3 CD : 57
F030 B3 E6 21 ED 79 38 03 21 : 7C
F038 02 00 22 A6 F0 3E 80 CD : 45
F040 E6 F0 38 44 CD 9B F0 07 : B1
F048 E6 DF 28 05 CD 31 EA 18 : F2
F050 37 D9 60 69 ED 4B 91 E0 : 82
F058 B7 ED 42 ED 4B 9F E0 B7 : 54
F060 ED 42 D9 28 08 3E 01 32 : A9
F068 8C E0 37 18 1B 2A 9F E0 : 7F
F070 ED 5B 91 E0 19 22 91 E0 : 65
F078 2A 93 E0 23 22 93 E0 01 : 56
SUM: 09 BA 55 3C 8A 42 26 07 EC0F

F080 FC 0F 3A 8B E0 ED 79 B7 : CD
F088 01 D0 1F 3A A2 E0 ED 79 : 12
F090 01 00 0B 3E 10 ED 79 FB : BB
F098 D1 C1 C9 ED 78 0F D0 0F : AE
F0A0 30 F9 4A ED 78 D9 ED 79 : 17
F0A8 03 D9 4B 18 EE F3 CD B3 : A0
F0B0 E6 21 ED 78 38 03 21 0A : D2
```

漢字変換エリアなどで使われる&HF000以降に、ディスクアクセスルーチンを置きました。「RWREC」ルーチンへのジャンプアドレスを変更しているの、かたちとしては「RWREC」の差し換えです。もうひとつは、「WTTRC」(1トラックフォーマットルーチン)の差し換えになります。この2つを代えるだけで動いてしまうのが、KAME-DOSのいいところでしょう。でも、内容は実は『試験に出るX1』そのまんまです。祝氏に感謝、感謝。

* * *

今月も長い解説でしたが、KAME-DOSの概要はわかってもらえたかねえ? とりあえずわからなくても、興味のあるところの知識が増えれば、それだけ将来役に立つこともあるでしょう。今月の記事は、むしろ他機種ユーザーでも、DOSに興味のある人に読んでほしいところです。DOSの構造を知る手掛かりになるだろうし、「システムプログラムとはこういうものだ!」というのもわかることでしょう。個々のルーチンのアルゴリズムも、知っておいて損はないと思いますよ。

それでは、来月はもっと使えるDOSにするために、外部コマンドの発表です。とりあえず来月まで、ばははい。に、

```
F0B8 00 22 DF F0 22 C0 F0 D9 : 9C
F0C0 ED 78 08 D9 3E A0 CD E6 : D7
F0C8 F0 DA 88 F0 CD D2 F0 C3 : 94
F0D0 47 F0 ED 78 0F D0 0F 30 : BA
F0D8 F9 4A 08 ED 79 D9 03 ED : 7A
F0E0 78 08 D9 4B 18 EC 32 8C : 66
F0E8 EB C5 D5 CD FC E6 DA 6F : 77
F0F0 E7 D1 C1 D8 01 F8 0F 3A : 93
F0F8 8C EB ED 79 3E 07 3D 20 : 7F
SUM: DB CA 6F F4 B0 44 9B 64 A2C0

F100 FD C9 C5 D5 CD 78 E8 D4 : 61
F108 FC E6 D4 4D E7 38 48 3E : A8
F110 03 32 5D F1 F3 CD B3 E6 : DC
F118 21 ED 78 38 03 21 0A 00 : EC
F120 22 DF F0 22 2F F1 16 FB : 44
F128 1E F8 D9 ED 4B 91 E0 ED : 85
F130 78 08 D9 01 F8 0F 3E 0F : 8F
F138 ED 79 3E 07 3D 20 FD CD : D2
F140 D2 F0 07 E6 FD 28 0C 21 : E3
F148 5D F1 35 C2 26 F1 CD 31 : 5A
F150 EA 18 04 CD 1F E7 B7 CD : 5D
F158 D9 E6 D1 C1 C9 00 : 1A
SUM: B4 05 5F 98 46 4F AE BC C67D
```

```
D01B C3 8C D7 25 #DIR3 JP DIR3
D01E C3 20 DA 26 #DPRE JP DPRE
D021 C3 F3 D3 27 #PREOP JP PREOPEN
D024 C3 CA D4 28 #OPENS JP OPENS
D027 C3 BB D5 29 #VDREV JP VDREV
D02A C3 28 DC 30 #DIRSB JP DIRSUB
D02D C3 85 DD 31 #SETUN JP SETUN
D030 C3 4A D2 32 #TRAN JP TRAN
D033 C3 A0 D1 33 #ALTA JP ALTA
D036 C3 D3 D1 34 #TRANR JP TRANRT
D039 35
D039 36 //////////////////////////////////////////////////
D039 37 : PROGRAM
D039 38 //////////////////////////////////////////////////
D039 39
D040 40 ORG $D860
D060 41
D060 42 // OPTION WORK /
D060 43 ;
D060 44 #ALPHA
D060 45 DS 26
D07A 46
D07A 47 // OPEN WORK /
D07A 48 ;
```

```
D080 49 ORG $D080
D080 50 ;
D080 51 #ZKUI DS 4
D080 52 #FNAM1 DS 46*1+35 F,M,D,M,C,DI
RN,C2
D13C 53
D13C 54 // WORK エリア 1 /
D13C 55 ;
D180 56 ORG $D180
D180 57 ;
D180 58 #CD DB 1 : FILE 番号
D180 59 #OP DB 0 : OPEN時 機能指示FLAG
D180 60 #FCRS DW 0 : クラスター数
D180 61 #YEN DB 0 : 文字列の大きさ
D180 62 #CDF DB 0 : コマンドがCDか?FLAG
D180 63 #FSZL DW 0 :
D180 64 #FSZH DW 0 : FILE-SIZE 4バイト
D180 65 #DRVR DB 7,7,7 : DEVICE-DRIVER
D180 66 構成
D180 67 #CTRL DB 32 : CTRL-CODE の上限
D180 68 // 階層DIR WORK /
D180 69 ;
D180 70 #CDIRN EQU $D800 : 階層の深さ
```



```

D42B D5      521 PUSH DE
D42C 22 8D E0 522 LD A,(#FF),HL;ADR SET
D42F 3A 97 E0 523 LD A,(#MAC)
D432 32 F7 DD 524 LD (SMAC),A
D435 CD 0C EC 525 CALL #DEVIC;自動判定
D438 E1      526 POP HL
D439 DD E1   527 POP IX
D43B 22 8D E0 528 LD A,(#FF),HL
D43E D1      529 POP DE
D43F C1      530 POP BC
D440 D8      531 RET C
D441 CD 83 D4 532 CALL CSDBM
D444 3A 8B E0 533 LD A,(#DN);OPENした
D447 DD 77 27 534 LD (IX+39),A;情報をSE
D44A 2A A4 E0 535 LD HL,(#BADR);T
D44D DD 75 25 536 LD (IX+37),L;
D450 DD 74 26 537 LD (IX+38),H;
D453 3A 97 E0 538 LD A,(#MAC);
D456 DD 77 28 539 LD (IX+40),A;
D459 DD E5   540 PUSH IX
D45B 3A 81 D1 541 LD A,(#OP);
D45E 6F      542 LD L,A;
D45F 3A 65 E0 543 LD A,(#SBD);
D462 67      544 LD H,A;
D463 E5      545 PUSH HL
D464 CD 8B D2 546 CALL KAISO;階層をCHECK
D467 E1      547 POP HL
D468 7C      548 LD A,H
D469 32 65 E0 549 LD (H+SBD),A
D46C 7D      550 LD A,L;
D470 32 81 D1 551 LD (#OP),A
D472 DD E1   552 POP IX
D473 3A 51 E0 553 RET C
D476 DD 77 2B 554 LD (IX+43),A
D479 2A 7C E0 555 LD HL,(#ECRS)
D47C DD 75 2C 556 LD (IX+44),L
D47F DD 74 2D 557 LD (IX+45),H
D482 C9      559 RET
D483          560
D484          561 ;(ROOT-DIRへもどす)
D485          562
D486          563 CSDBM
D487 3A F7 DD 564 LD A,(SMAC)
D488 21 97 E0 565 LD HL,#MAC
D489 BE      566 CP (HL)
D48A C8      567 RET Z;IF 開しFORMAT RET
D48B 21 00 F0 568 LD HL,CSDBM
D48C CD 88 DD 569 CALL DNST2
D491 AF      570 XOR A,ROOTへ
D492 77      571 LD (HL),A
D493 C9      572 RET
D494          573
D495          574 ;(DEVICE-DRIVER 欄分付)
D496          575
D497          576 HDRV
D498 78      577 LD A,B
D499 FE 03    578 CP 3
D49F C9      579 RET NZ IF 3文字以外 RET
D49F C9      580 XOR A
D499 32 84 D1 581 LD (#YEN),A
D49E 21 40 EC 582 LD HL,SRDR;TABLE
D49F 0E 00    583 LD C,0
D4A1          584 HDRVLP2
D4A1 D5      585 PUSH DE
D4A2 E5      586 PUSH HL
D4A3 0E 03    587 LD B,3;3文字
D4A5          588 HDRVLP
D4A5 1A      589 LD A,(DE)
D4A6 BE      590 CP (HL)
D4A7 28 06    591 JR Z,HDRVSK
D4A9 CD F7 D8 592 CALL OKCHG
D4AC BE      593 CP (HL)
D4AD 20 0F    594 XOR NZ,HDRVSK
D4AF          595 HDRVSK
D4B1          596 INC DE
D4B2 13      597 INC HL
D4B3 10 F2    598 DN2 HDRVLP;DRIVER 指定
D4B4 21 8A D1 600 LD HL,#DVR;WORK TABLE
D4B7 51      601 LD D,C;識別CODE
D4B8 CD 6C D7 602 CALL ODST
D4B9 72      603 LD (HL),D
D4BC D1      604 POP DE
D4BD C9      605 RET
D4BE          606 HDRVSK
D4BF 0C      607 INC C
D4BF E1      608 POP HL
D4C0 11 06 00 609 LD DE,6;6 SKIP
D4C3 19      610 ADD HL,DE
D4C4 D1      611 POP DE
D4C5 7E      612 LD A,(HL)
D4C6 B7      613 OR A
D4C7 20 D8    614 JR NZ,HDRVLP2
D4C9 C9      615 RET
D4CA          616 ;OPEN 処理 2 /
D4CA          617
D4CA          618
D4CA          619 OPENS IN (DE)
D4CB CD 21 D6 620 CALL NOTHR;NAME SET
D4CD DD      621 RET C
D4CE CD 76 D7 622 CALL ODST2
D4D1 E5      623 PUSH HL
D4D2 DD E1    624 POP IX
D4D4 DD 6E 23 625 LD L,(IX+35);以下
D4D7 DD 66 24 626 LD L,(IX+36);情報設定
D4DA 22 8D E0 627 LD A,(#FF),HL
D4DD DD 6E 25 628 LD L,(IX+37);
D4DE DD 66 26 629 LD L,(IX+38);
D4E3 22 8D E0 630 LD (#MORF),HL
D4E6 DD 7E 27 631 LD A,(IX+39);
D4E9 32 8B E0 632 LD (#DN),A
D4EC DD 7E 28 633 LD A,(IX+40);
D4EF 32 97 E0 634 LD (#MAC),A
D4F2 CD 1B DD 635 CALL #VAR
D4F5 CD 1E DD 636 CALL #FREE
D4F8 2A 95 E0 637 LD HL,(#CRS);
D4FB 22 8D D1 638 LD (#CRS),HL
D4FE DD 7E 2B 639 LD A,(IX+43);
D501 32 51 E0 640 LD (#DIRN),A
D504 DD 6E 2C 641 LD L,(IX+44);
D507 DD 66 2D 642 LD H,(IX+45);
D50A 22 95 E0 643 LD (IX+CRS),H
D50D DD E5   644 PUSH IX
D50F CD 44 D5 645 CALL OPN2
D512 DD E1    646 POP IX
D514 DD      647 RET C
D515 CD 76 D7 648 CALL ODST2
D51B BE      649 EX DE,HL
D519 21 68 E0 650 LD HL,#FNAM
D51C 01 23 00 651 LD BC,35
D51F DD B0    652 LDIR
D521 2A 7C E0 653 LD HL,(#ECRS)
D524 DD 75 29 654 LD (IX+41),L
D527 DD 74 2A 655 LD (IX+42),L
D52A DD 75 14 656 LD (IX+20),L
D52D DD 74 15 657 LD (IX+21),H
D530 2A 52 E0 658 LD HL,(#MORF)
D533 DD 75 2E 659 LD (IX+37),L
D536 DD 74 26 660 LD (IX+38),H
D539 21 8C DD 661 LD H,#ZOKU1
D53C CD 60 D7 662 CALL ODST
D53F 3A A3 E0 663 LD A,(#ZOKU)
D542 77      664 LD (HL),A
D543 C9      665 RET
D544          666
D545          667 OPN2
D546 3A 81 D1 668 LD A,(#OP)
D547 FE 03    669 CP 3
D549 DA 00 D0 670 JP C,#DIR
D54C 28 05    671 JR Z,OPNSK2
D54E CD 00 D0 672 CALL #DIR
D551 18 03    673 JR OPNSK3
D553          674 OPNSK2
D555          675 CALL #DIR3
D556          676 OPNSK3
D558 30 09    677 JR NC,OPNSK1
D55A 3A 8C E0 678 LD A,(#STOP)
D55B FE 01    679 CP 3
D55D 28 09    680 JR Z,OPNSK4
D55F 37      681 SCF
D560 C9      682 RET
D561          683 OPNSK1;WRITE
D563 3A 89 E0 684 LD A,(#FOPD+2)
D564 B7      685 OR A
D565 C2 42 E0 686 JP NZ,#ERR7
D568          687 OPNSK4
D568 AF      688 XOR A
D569 32 8C E0 689 LD (#STOP),A
D56C 21 01 00 690 LD HL,1
D56F 22 95 E0 691 LD (#CRS),HL
D572 CD 21 E0 692 CALL #SEC00;空をさがす
D575 2A 95 E0 693 LD HL,(#CRS)
D578 22 7C E0 694 LD (#ECRS),HL
D57B B7      695 OR A
D57C C9      696 RET
D57D          697
D57E          698 ;データ入出力決定 /
D57F          699
D580          700 ICPO
D581          701 LD A,(#IOFG)
D582          702 DEC A;1:人力要求
D583 28 2D    703 JR Z,ICPO2
D584 28 13    704 DEC A;2:出力要求
D585 28 13    705 JR Z,ICPENT1
D586          706 ICPO2;10:初期化
D587 3E 01    707 LD L,1
D588 2A 80 D1 708 LD (#OD),A
D589 2A A4 E0 709 LD HL,(#BADR)
D58E 22 91 E0 710 LD (#FF),HL;ADR
D591 FD 21 98 D5 711 LD IX,ICPO2;
D595 C3 BB D5 712 JP DEVDRV;
D598          713 ICPO2;=CALL DEVDRV
D599          714 RET C;
D59A          715 ICPENT1
D59B          716 CALL #MSX1;変換
D59C 3A 9C E0 717 LD A,(#IOFG)
D59F FE 02    718 CP 2
D5A1 28 13    719 JR Z,ICPOLP
D5A3 3E 02    720 LD A,2
D5A5 32 80 D1 721 LD (#OD),A
D5A8 FD 21 AF D5 722 LD IX,ICPO3;
D5AC C3 BB D5 723 JP DEVDRV;
D5AF          724 ICPO3;=CALL DEVDRV
D5B0          725 RET C;
D5B1          726 ICPO2
D5B2 3A 9C E0 727 LD A,(#IOFG)
D5B3 B7      728 OR A
D5B4 20 E3    729 JR NZ,ICPENT1
D5B6 AF      730 XOR A
D5B7 32 80 D1 731 LD (#OD),A
D5B8 C9      732 RET
D5BB          733
D5BB          734 DEVDRV;変換のJUMP
D5BD 21 8A D1 735 LD HL,HDRV
D5BE CD 6C D7 736 CALL ODST
D5C1 7E      737 LD A,(HL)
D5C2 87      738 ADD A,A
D5C3 4F      739 LD C,A
D5C4 87      740 ADD A,A
D5C5 81      741 ADD A,C
D5C6 DD 21 40 EC 742 LD IX,SRDR
D5CA 4F      743 LD C,A
D5CB 06 00    744 LD B,0
D5CC DD 09    745 ADD IX,BC
D5CD DD 7E 03 746 LD A,(IX+3)
D5D2 B7      747 OR A
D5D3 C8      748 RET Z;0: BASIC
D5D4 DD 6E 04 749 LD L,(IX+4)
D5D7 DD 66 05 750 LD H,(IX+5)
D5DA FD E5    751 PUSH IX;
D5DC E9      752 JP (HL);CALL (HL)
D5DD          753
D5DD          754 DEVOT;IN BF DISK専門
D5DD 3A 80 D1 755 LD A,(#OD)
D5E0 30      756 DEC A
D5E1 32 A1 E0 757 LD (#FRWF),A;RAW FLUG
D5E4 CD 76 D7 758 CALL ODST2
D5E7 E5      759 PUSH HL
D5E8 DD E1    760 POP IX
D5EA DD 6E 23 761 LD L,(IX+35);以下
D5ED DD 66 24 762 LD H,(IX+36);情報設定
D5F0 22 8D E0 763 LD A,(#FF),HL
D5F3 DD 6E 29 764 LD L,(IX+41);
D5F6 DD 66 2A 765 LD H,(IX+42);
D5F9 22 95 E0 766 LD (#CRS),HL
D5FC DD 7E 28 767 LD A,(IX+40);
D5FF 32 97 E0 768 LD (#MAC),A
D602 DD 7E 27 769 LD A,(IX+39);
D605 32 8B E0 770 LD (#DN),A
D608 DD 1B E0 771 PUSH IX
D60A CD 09 E0 772 CALL #VAR
D60B CD 1E DD 773 CALL #FREE;RAW
D610 DD E1    774 POP IX
D612 DD      775 RET C
D613 2A 95 E0 776 LD HL,(#CRS)
D616 DD 75 29 777 LD (IX+41),L
D619 DD 74 2A 778 LD (IX+42),H
D61C          779 NULRET
D61F AF      780 XOR A
D620 32 A1 E0 781 LD (#FRWF),A
D622          782 RET
D623          783 ;(FILE-NAME 展開)
D624          784
D625          785
D626 20 19    786 MOTH2 IN (DE)
D628 3A 81 D1 787 CALL CLRFRM;MEM CLEAR
D62B B7      788 LD A,B
D625 B7      789 XOR A
D626 20 19    790 JR NZ,MOTH2SK2;IF 長さ>0
D628 3A 81 D1 791 LD A,(#OP);1:長さを
D62B B7      792 OR A
D62C 3A 36 E0 793 JP Z,#ERR3
D62F FE 02    794 CP 2
D631 3A 36 E0 795 JP Z,#ERR3
D634 FE 03    796 CP 3
D636 C8      797 RET Z
D637 3E 2A    798 LD A,'*'
D639 32 CD DD 799 LD (BGAM),A
D63C 32 D4 DD 800 LD (BGAM+13),A
D63F 18 03    801 JR MOTH2SK3
D641          802 MOTH2SK2
D641 CD 7B D6 803 CALL INSTEN
D644 01 10 00 804 MOTH2SK3
D644 DD BC,16 805 LD BC,16
D647 21 C7 DD 806 LD HL,BGAM
D64A 3A 81 D1 807 LD A,(#OP)
D64F 28 07    808 CP 3
D651 11 54 E0 809 LD DE,#FLNM
D654 DD E0    810 LDIR
D656 B7      811 OR A
D657 C9      812 RET
D658          813
D658          814 MOTH2SK4
D658          815 LD B,13
D659 DD 6E 00 816 LD DE,#FNAM
D65A 11 68 E0 817 LD DE,SUGO2
D65D CD 6D D6 818 CALL SUGO2
D660 0E 03    819 LD B,3
D662 21 D4 DD 820 LD HL,BGAM+13
D665 11 75 E0 821 LD DE,#FNAM+13
D668 CD 6D D6 822 CALL SUGO2
D66B B7      823 OR A
D66C C9      824 RET
D66D          825 SUGO2
D66D 7E      826 LD A,(HL)
D66E FE 2A    827 CP '+'
D66F C8      828 RET Z
D671 FE 3F    829 CP '?'
D673 28 01    830 JR Z,SUGOS
D675 12      831 LD (DE),A
D676          832 SUGOS
D676 23      833 INC HL
D677 13      834 INC DE
D678 10 F3    835 DJNZ SUGO2
D67A C9      836 RET
D67B          837
D67B          838 INSTEN;以下 NAME 処理
D67B D5      839 PUSH DE
D67C 48      840 LD C,B;C=DATA LENGTH
D67D 0E 0D    841 LD B,13
D67F 21 C7 DD 842 LD HL,BGAM
D682 1A      843 INSTENLP
D683 13      844 LD A,(DE)
D684 FE 2E    845 INC DE
D685          846 CP '+'
D686 28 19    847 JR Z,INSTENSK
D688 77      848 LD (HL),A;DATA SET
D689 23      849 INC HL
D68A DD      850 DEC C
D68B 28 07    851 JR Z,INSTENSK2
D68D 10 F3    852 DJNZ INSTENLP
D68F CD B4 D6 853 CALL INSTEN2;FIND '+'
D692 30 0D    854 JR NC,INSTENSK;IF あれば JR
D694          855 INSTENSK2; '+'がない
D694 D1      856 POP DE
D695 3A 81 D1 857 LD A,(#OP);
D698 FE 02    858 CP 2 ;(#OP)=2:拡張子='*'
D699          859 RET NZ;
D69B 3E 2A    860 LD A,'*'
D69D 32 D4 DD 861 LD (BGAM+13),A
D6A0 C9      862 RET
D6A1          863
D6A1          864 INSTENSK;拡張子処理
D6A1          865 DEC C
D6A2 28 0E    866 JR Z,INSTSS
D6A4 0E 03    867 LD B,3
D6A5 21 D4 DD 868 LD HL,BGAM+13
D6A9          869 INSTLP2
D6AA 1A      870 LD A,(DE)
D6AB 77      871 LD (HL),A
D6AB 13      872 INC DE
D6AC 23      873 INC HL
D6AD 0D      874 DEC C
D6AE 28 02    875 JR Z,INSTSS
D6B0 10 F7    876 DJNZ INSTLP2
D6B2          877 INSTSS
D6B3 D1      878 POP DE
D6B3 C9      879 RET
D6B4          880
D6B4          881 INSTEN2;OVER 13文字
D6B4 1A      882 LD A,(DE)
D6B5 13      883 INC DE
D6B6 FE 2E    884 CP '+'
D6B8 C8      885 PUSH DE
D6B9 DD      886 DEC C
D6BA 20 F8    887 JR NZ,INSTEN2
D6BC 37      888 SCF
D6BD C9      889 RET
D6BE          890
D6BE          891 CLRFRM;WORK
D6BF D5      892 PUSH BC;初期化
D6C0 FE      893 PUSH DE
D6C1 01 0F 00 894 PUSH HL
D6C2 3E 20    895 LD BC,15
D6C3 3E 20    896 LD A,'*'
D6C4 11 C8 DD 897 LD DE,BGAM+1
D6C5 21 C7 DD 898 LD HL,BGAM
D6C6 77      899 LD (HL),A
D6C7 DD      900 LDIR
D6C8 E1      901 POP HL;#BADR
D6C9 D1      902 POP DE
D6D1 C1      903 POP BC
D6D2 C9      904 RET
D6D3          905
D6D3          906 ;CLOSE 処理 /
D6D3          907
D6D3          908 CLOSE;IN FNAM,FSZL
D6D3 CD 76 D7 909 CALL ODST2
D6D6 E5      910 PUSH HL;
D6D7 DD 81    911 POP IX;LD IX,HL
D6D9 2A 86 D1 912 LD HL,(#FSZL);以下
D6DB DD 75 18 913 LD (IX+19),L;情報SET
D6DE DD 74 11 914 LD (IX+17),H
D6E0 2A 88 D1 915 LD HL,(#FSZH)
D6E5 DD 75 12 916 LD (IX+18),L
D6E8 DD 74 13 917 LD (IX+19),H
D6EB 2A A4 E0 918 LD HL,(#BADR)
D6EE 22 82 E0 919 LD (#MORF),H
D6F1 DD 7E 2B 920 LD A,(IX+43)
D6F4 32 51 E0 921 LD (#DIRN),A
D6F7 DD 7E 2C 922 LD L,(IX+44)
D6FA DD 66 2D 923 LD H,(IX+45)
D6FD DD 95 E0 924 LD (#CRS),H
D700 CD 1B DD 925 JP C,#ERR3
D703 2A A4 E0 926 LD HL,#ODST2
D706 D4 36 D7 927 CALL NC,CLOSESK
D709 DD 52 E0 928 LD (#MORF),HL
D70C AF      929 XOR A
D70D 32 8C E0 930 LD (#STOP),A
D710 CD D4 D9 931 CALL DSECO;空DIRをさがす
D713 DA 3C E0 932 JP C,#ERR5
D716 CD 76 D7 933 CALL #ODST2
D719 11 68 E0 934 LD DE,#FNAM
D71C 01 23 00 935 LD BC,35
D71F DD B0    936 LDIR
D720 11 80 DD 937 LD HL,#ZOKU1
D724 CD 6C D7 938 CALL ODST
D727 7E      939 LD A,(HL)
D728 32 A3 E0 940 LD (#ZOKU),A;属性
D72B 3E 01    941 LD A,1;
D72D 32 A1 E0 942 LD (#FRWF),A;WRITE
D730 CD 07 D9 943 CALL NAGRM;NAME SET
D733 C3 3C D7 944 JP CLOSE2
D736          945
D736          946 CLOSESK
D736 CD 03 E0 947 CALL #DLFAT;DEL FAT
D739 C3 12 DD 948 JP #DLDIR;DEL NAME
D73C          949
D73C          950 ;DISK CLOSE /
D73C          951
D73C          952 CLOSE2
D73C CD 76 D7 953 CALL ODST2
D73F E5      954 PUSH HL;
D740 DD E1    955 POP IX;LD IX,HL
D742 DD 7E 2B 956 LD A,(IX+43)
D745 32 51 E0 957 LD L,(IX+44)
D748 B7      958 OR A
D749 28 11    959 JR Z,CLOSE2SK;ROOT
D74B DD 6E 2C 960 LD L,(IX+44)
D74D DD 6E 2D 961 LD L,(IX+45)
D751 22 7C E0 962 LD (#ECRS),HL
D754 E5      963 PUSH HL;DIRがはみだす
D755 CD 03 E0 964 CALL #DLFAT;DEL FAT
D758 E1      965 POP HL;
D759 DD 22 95 E0 966 LD (#CRS),H
D75C          967 CLOSE2SK
D75F DD 6E 23 968 LD L,(IX+35)
D75F DD 6E 24 969 LD H,(IX+36)
D762 22 8D E0 970 LD (#FF),HL

```

▶5月12日、ダンジョンマスターをやっていると、雨が降り始めた。そのうち雨にまじって白いものも降ってきた。霰かなと思っていると、つぶが大きくなってきたので霰だとわかった。おかけて、学校のガラスが500枚以上割れ14日が休みに。ピンポン玉ぐらいのものもあったそうで、まさに、天災は忘れた頃にやってくる。

佐藤 史彦 (17) 北海道


```

D765 CD 03 D0 971 CALL #DIRWT;WRITE
D766 D4 06 E0 972 CALL NC,#FATWT;
D768 C9 973 RET
D76C 974
D76C 975 ;/HL=HL+(#OD)/
D76C 976
D76C 977 ODST;IN HL OUT HL
D76C C5 978 PUSH BC
D76D 3A 80 D1 979 LD A,(#OD)
D770 4F 980 LD C,A
D771 06 00 981 LD B,0
D773 09 982 ADD HL,BC
D774 C1 983 POP BC
D775 C9 984 RET
D776 985
D776 986 ;/HL=HL+46*(#OD)/
D776 987 ;46..FNAM1の大きさ
D776 988 ODST2
D776 3A 80 D1 989 LD A,(#OD)
D779 6F 990 LD L,A
D77A 26 00 991 LD H,0
D77C 11 2E 00 992 LD DE,46
D77D CD 27 E0 993 CALL #MULT
D782 11 84 D0 994 LD DE,#FNAM1
D785 19 995 ADD HL,DE
D786 C9 996 RET
D787 997
D787 998 ;/READ DIR/
D787 999
D787 1000 DIR;IN FLNM
D787 21 54 E0 1001 LD HL,#FLNM
D78A 16 03 1002 JR DIR3SK
D78C 1003 DIR3
D78C 21 68 E0 1004 LD HL,#FNAM ;FLNM保存時
D78F C5 1005 PUSH BC
D78F 22 EA DD 1006 LD (BFNM),HL
D792 CD B7 D7 1007 CALL EXTND ;'-'-'-'
D795 AF 1008 XOR A
D796 32 A1 E0 1009 LD (#FRWF),A;READ
D799 CD D8 D7 1010 CALL DIRRW
D79C 2A 91 E0 1011 LD HL,(#BF)
D79F 22 20 DE 1012 LD (SHIME),HL
D7A2 C3 2F D8 1013 JP DIR2 ;FIND NAME
D7A5 1014
D7A5 1015 ;/WRITE DIR/
D7A5 1016
D7A5 1017 DIRWT
D7A5 3E 01 1018 LD A,1 ;WRITE
D7A7 32 A1 E0 1019 LD (#FRWF),A;
D7AA CD D8 D7 1020 CALL DIRRW
D7AD 38 20 1021 LD A,32
D7AF 32 98 E0 1022 LD (#PASS),A;PASSWORD
D7B2 AF 1023 XOR A
D7B3 32 A1 E0 1024 LD (#FRWF),A
D7B6 C9 1025 RET
D7B7 1026
D7B7 1027 ;/"*"を"?"に展開/
D7B7 1028
D7B7 1029 EXTND;IN FLNM OUT FLNM
D7B7 C5 1030 PUSH BC
D7B8 D5 1031 PUSH DE
D7B9 E5 1032 PUSH HL
D7BA 2A EA DD 1033 LD HL,(BFNM)
D7BD 0E 02 1034 LD C,2
D7BF 06 0D 1035 LD B,13
D7C1 7E 1036 EXTNDLP1
D7C1 7E 1037 LD A,(HL)
D7C2 FE 2A 1038 CP #?
D7C4 28 05 1039 JR Z,EXTNDSK1
D7C6 23 1040 INC HL
D7C7 10 F8 1041 DJNZ EXTNDLP1
D7C9 18 06 1042 JR EXTNDSK2
D7CB 1043 EXTNDSK1
D7CB 3E 3F 1044 LD A,?'
D7CD 7F 1045 LD (HL),A
D7CE 23 1046 INC HL
D7CF 10 FA 1047 DJNZ EXTNDSK1
D7D1 1048 EXTNDSK2;拡張子
D7D1 06 03 1049 LD B,3
D7D3 0D 1050 DEC C
D7D4 20 EB 1051 JR NZ,EXTNDLP1
D7D6 B7 1052 OR A
D7D7 E1 1053 POP HL
D7D8 D1 1054 POP DE
D7D9 C1 1055 POP BC
D7DA C9 1056 RET
D7DB 1057
D7DB 1058 ;/READ & WRITE DIR/
D7DB 1059
D7DB 1060 DIRRW;IN BSIZ,DIRN,CRS
D7DB 2A A4 E0 1061 LD HL,(#BADR)
D7DB 22 91 E0 1062 LD (#BF),HL
D7E1 3A 51 E0 1063 LD A,(#DIRN)
D7E4 B7 1064 OR A
D7E5 C2 09 E0 1065 JP NZ,#CRSRW;障害時
D7E8 C5 1066 PUSH BC
D7E9 CD F8 D7 1067 CALL RDIRD;FORMAT割設定
D7EC 1068 RTDIRLP
D7EC CD 00 E0 1069 CALL #RWREC;RAW 1レコード
D7EF 38 05 1070 JR C,RDLPXET
D7F1 0D 1071 DEC C ;DIRレコード長
D7F2 28 02 1072 JR Z,RDLPXET
D7F4 10 F6 1073 DJNZ RTDIRLP
D7F6 1074 RDLPXET
D7F6 C1 1075 POP BC
D7F7 C9 1076 RET
D7F8 1077
D7F8 1078 RDIRD
D7F8 D5 1079 PUSH DE
D7F9 E5 1080 PUSH HL
D7FA 3A 97 E0 1081 LD A,(#MAC);FORMAT
D7FD DD 21 34 E1 1082 LD IX,RODT1
D801 3D 1083 DEC A;1..X12HD
D802 28 12 1084 JR Z,RDIRDSK
D804 DD 21 36 E1 1085 LD IX,RODT2
D808 3D 1086 DEC A;2..MS-DOS2HD
D809 28 0B 1087 JR Z,RDIRDSK
D80B DD 21 38 E1 1088 LD IX,RODT3
D80F 3D 1089 DEC A;3..X12D
D810 28 04 1090 JR Z,RDIRDSK
D812 DD 21 3A E1 1091 LD IX,RODT4;4..MS-DOS32D
D816 1092 RDIRD
D816 DD 0E 00 1093 LD L,(IX+0)
D819 26 00 1094 LD H,0
D81B 22 93 E0 1095 LD (RREC),HL;レコード
D81E 2A 8F E0 1096 LD HL,(#BSIZ)
D821 ED 5B 9F E0 1097 LD DE,(#SLNG)
D825 CD 2A E0 1098 CALL #DIVHD
D828 47 1099 LD B,A
D829 DD 4E 01 1100 LD C,(IX+1);レコード長
D82C E1 1101 POP HL
D82D D1 1102 POP DE
D82E C9 1103 RET
D82F 1104
D82F 1105 ;/展開するFILE-NAMEをさがす/
D82F 1106
D82F 1107 DIR2;IN MORF OUT MORF,DBL
D82F CD 38 D8 1108 CALL SFLNM
D832 CA 07 D9 1109 JP Z,NAGRM;NAME SET
D835 C3 06 E0 1110 JP #ERR3 ;一致なし
D838 1111
D838 1112 ;/FILE-NAMEが一致するか/
D838 1113
D838 1114 SFLNM;IN MORF OUT MORF
D838 DD E5 1115 PUSH IX
D83A CD 53 DD 1116 CALL SETFNDT;FORMAT割設定
D83D 06 00 1117 LD B,128;ファイルは128まで
D83F 2A 52 E0 1118 LD HL,(#MORF)
D842 1119 SFLNMLP
D842 ED 5B 20 DE 1120 LD DE,(SHIME);DIRの最後

D846 B7 1121
D847 ED 52 1122 SRC HL,DE
D849 30 28 1123 JR NC,SFLD;
D84B 19 1124 ADD HL,DE;=LD HL,(#MORF)
D84C CD 12 E0 1125 CALL #LDAHL
D84F DD BE 00 1126 CP (IX+0);END
D852 28 1F 1127 JR Z,SFLD
D854 DD BE 01 1128 CP (IX+1);KILL
D857 28 14 1129 JR Z,NMKS
D859 C5 1130 PUSH BC
D85A E5 1131 PUSH HL
D85B CD A5 D8 1132 CALL DZGT;属性
D85E CD B0 D8 1133 CALL SKDIR;SKIP?
D861 CD C2 D8 1134 CALL Z,SKDIR2;SKIP?
D864 F1 1135 POP HL
D865 E5 1136 PUSH HL
D866 CD 7E D8 1137 CALL Z,NIDFY;NAME?
D869 E1 1138 POP HL
D86A C1 1139 POP BC
D86B 28 0B 1140 JR Z,SFLNMSK;発見
D86D 1141 NMKS
D86D 11 20 00 1142 LD DE,32;NEXT
D86F 19 1143 ADD HL,DE;
D871 10 CF 1144 DJNZ SFLNMLP;
D873 1145 SFLD
D874 3F 1146 XOR A
D874 3C 1147 INC A;RZF
D875 2A 52 E0 1148 LD HL,(#MORF)
D878 1149 SFLNMSK
D878 22 52 E0 1150 LD (#MORF),HL
D87B DD E1 1151 POP IX
D87D C9 1152 RET
D87E 1153
D87E 1154 NIDFY;IN HL OUT Z FLAG
D87E DD 5E 02 1155 LD E,(IX+2)
D881 16 00 1156 LD D,0
D883 19 1157 ADD HL,DE
D884 CD 12 E0 1158 CALL #LDAHL
D887 D6 2E 1159 SUB
D889 20 02 1160 JR NZ,NIDBSK
D88B 3C 1161 INC A
D88C C9 1162 RET
D88D 1163 NIDBSK
D88D ED 5B EA DD 1164 LD DE,(BFNM)
D891 EB 1165 EX DE,HL;HL=#FLNM,DE=BUFF
D892 DD 46 03 1166 LD B,(IX+3);B=LENGTH
D895 CD E1 D8 1167 CALL NIDLP;NAME?
D898 C9 1168 INC A
D899 2A EA DD 1169 LD HL,(BFNM)
D89C 01 0D 00 1170 LD BC,13
D89F 69 1171 ADD HL,BC
D8A0 06 03 1172 LD B,0;B=LENGTH
D8A2 C3 E1 D8 1173 JP NIDLP;NAME?
D8A5 1174
D8A5 1175 DZGT
D8A5 DD 5E 04 1176 LD E,(IX+4)
D8A8 16 00 1177 LD D,0
D8AA 19 1178 ADD HL,DE
D8AB CD 12 E0 1179 CALL #LDAHL
D8AE 47 1180 LD B,A
D8AF C9 1181 RET
D8B0 1182
D8B0 1183 SKDIR
D8B0 3A 65 E0 1184 LD A,(#SDDR)
D8B3 B7 1185 OR A
D8B4 C8 1186 RET Z;0
D8B5 DD 4E 07 1187 LD C,(IX+7)
D8B8 3D 1188 RET
D8B9 28 04 1189 JR Z,SKDIR11
D8BB 78 1190 LD A,B
D8BC A1 1191 AND C
D8BD A9 1192 XOR C
D8BE C9 1193 RET;D2
D8BF 78 1194 SKDIR11
D8BF 78 1195 LD A,B
D8C0 A1 1196 AND C
D8C1 C9 1197 RET;D1
D8C2 1198
D8C2 1199 SKDIR2
D8C2 3A 64 E0 1200 LD A,(#DPNF)
D8C5 B7 1201 OR A
D8C6 C8 1202 RET Z;0
D8C7 FE 02 1203 CP 2
D8C9 28 04 1204 JR Z,SKDIR22
D8CB FE 03 1205 CP 3
D8CD 28 0A 1206 JR Z,SKDIR23
D8CF 78 1207 LD A,B
D8D0 DD A6 05 1208 AND (IX+5);S1
D8D3 C9 1209 RET;NZ
D8D4 1210 SKDIR22
D8D4 78 1211 LD A,B
D8D5 DD A6 06 1212 AND (IX+6);V12
D8D8 C9 1213 RET
D8D9 1214 SKDIR23
D8DE DD A6 06 1215 LD A,B
D8DE DD A6 06 1216 AND (IX+6)
D8E0 DD A6 06 1217 XOR (IX+6)
D8E2 C9 1218 RET
D8E1 1219
D8E1 1220 NIDLP;展開NAME?
D8E1 7E 1221 LD A,1
D8E2 FE 3F 1222 CP '?'
D8E4 28 0B 1223 JR Z,NIDWILD
D8E6 CD 15 E0 1224 CALL #LDADE
D8E8 BE 1225 CP (HL)
D8EA 28 05 1226 JR Z,NIDWILD;大小交換
D8EC CD F7 D8 1227 CALL OKCHG;大小交換
D8EE BE 1228 CP (HL)
D8F0 C8 1229 NET NZ;NZ=違う
D8F1 1230 NIDWILD
D8F1 23 1231 INC HL
D8F2 13 1232 INC DE
D8F3 10 EC 1233 DJNZ NIDLP
D8F5 AF 1234 XOR A;Z=展開する
D8F6 C9 1235 RET
D8F7 1236
D8F7 1237 ;/大文字 & 小文字変換/
D8F7 1238 A->a a->A
D8F7 E5 1239 OKCHG;IN A OUT A
D8F9 67 1240 PUSH HL
D8F9 E6 20 1241 LD H,A
D8F9 E6 20 1242 AND #20
D8FB 20 05 1243 JR NZ,OKCHGS
D8FD 3E 20 1244 LD A,#20;A&00100000
D8FF B4 1245 OR H;大->小
D900 18 03 1246 JR OKCHGRT
D902 1247
D902 1248 LD A,BDF;A&10111111
D904 A4 1249 AND H;小->大
D905 1250 OKCHGRT
D905 E1 1251 POP HL
D906 C9 1252 RET
D907 1253
D907 1254 ;/FILE情報をWORKにSET/
D907 1255
D907 1256 NAGRM;IN MORF OUT DBL
D907 CD 96 D9 1257 CALL FMSSETS;CLEAR
D90A 3A 97 E0 1258 LD A,(#MAC);FORMAT割
D90D 21 4C E1 1259 LD HL,NADT1
D910 3D 1261 DEC A
D911 28 0F 1262 JR Z,NASKMK
D913 21 72 E1 1263 LD HL,NADT2
D916 3D 1264 DEC A
D917 28 09 1265 JR Z,NASKMK
D919 21 4C E1 1266 LD HL,NADT1
D91C 3D 1267 DEC A
D91D 28 03 1268 JR Z,NASKMK
D91F 21 72 E1 1269 LD HL,NADT2
D922 1270 NASKMK

D922 1271
D922 46 1272 LD B,(HL);SET DATA数
D923 23 1273 INC HL;
D924 1274 NAGRLP2
D924 C5 1275 PUSH BC
D925 4E 1276 LD C,(HL);BC=位置
D926 06 00 1277 LD B,0
D928 23 1278 INC HL
D929 7E 1279 LD A,(HL);A=長さ
D92A 23 1280 INC HL
D92D 5E 1281 INC E,(HL);DE=
D92C 23 1282 INC HL;SETするADR
D92D 56 1283 LD D,(HL);
D92E 23 1284 INC HL
D92E E5 1285 PUSH HL
D930 2A 52 E0 1286 LD HL,(#MORF)
D933 09 1287 ADD HL,BC;(HL)から
D934 47 1288 LD B,A;(DE)へ
D935 CD B5 D9 1289 CALL JITS;B図 SET
D938 E1 1290 POP HL
D939 C1 1291 POP BC
D93A 10 E8 1292 DJNZ NAGRLP2
D93C CD 48 D9 1293 CALL NADIN;属性
D93F 2A 52 E0 1294 LD HL,(#MORF)
D942 11 20 00 1295 LD DE,32;
D945 19 1296 ADD HL,DE;NEXT
D946 22 52 E0 1297 LD HL,(#MORF);HL
D949 B7 1298 OR A
D94A C9 1299 RET
D94B 1300
D94B 1301 ;/FILE属性をWORKにSET/
D94B 1302
D94B 1303 NADIN
D94B 4E 1304 LD C,(HL)
D94C 23 1305 INC HL
D94D 06 00 1306 LD B,0
D94F EB 1307 EX DE,HL
D950 2A 52 E0 1308 LD HL,(#MORF)
D953 09 1309 ADD HL,BC
D954 3A A1 E0 1310 LD A,(#FRWF);
D957 B7 1311 OR A
D958 20 15 1312 JR NZ,NADIN2;WRITE
D95A CD 12 E0 1313 CALL #LDAHL
D95D 4F 1314 LD C,A;BIT位置
D95E 21 87 E0 1315 LD HL,#FDOOP;WORK
D961 06 04 1316 LD B,4;4バイト
D963 1317 NADINLP
D963 1A 1318 LD A,(DE)
D964 13 1319 INC DE
D965 A1 1320 AND C;BIT取出
D966 77 1321 LD (HL),A
D967 23 1322 INC HL
D968 10 F9 1323 DJNZ NADINLP
D96A 79 1324 LD A,C;属性そのもの
D96B 32 A3 E0 1325 LD (#ZOKU),A
D96E C9 1326 RET
D96F 1327
D96F 1328 NADIN2;WRITE
D96F E5 1329 PUSH HL;MORF+
D970 21 87 E0 1330 LD HL,#FDOOP
D973 EB 1331 EX DE,HL
D974 06 04 1332 LD B,4
D976 0E 00 1333 LD C,0
D978 1A 1334 NADINLP2
D978 1A 1335 LD A,(DE);FDOOP
D979 B7 1336 OR A
D97A 28 03 1337 JR Z,NADINSK2
D97C 78 1338 LD A,C
D97D B6 1339 OR (HL);C<C OR (HL)
D97E 4F 1340 LD C,A;
D97F 77 1341 NADINSK2
D97F 13 1342 INC DE
D980 23 1343 INC HL
D981 10 F5 1344 DJNZ NADINLP2
D983 D1 1345 POP DE;MORF+
D984 CD B0 D9 1346 CALL ZOKUS
D987 79 1347 LD A,C
D988 C3 18 E0 1348 JP #LDDA
D98B 1349
D98B 1350 ZOKUS
D98B 3A A3 E0 1351 LD A,(#ZOKU)
D98E B7 1352 OR A
D98F C8 1353 RET Z
D990 4F 1354 LD C,A
D991 AF 1355 XOR A
D992 32 A3 E0 1356 LD (#ZOKU),A
D995 C9 1357 RET
D996 1358
D996 1359 ;/エリアをCLEAR/
D996 1360
D996 1361 FMSSETS
D996 3A A1 E0 1362 LD A,(#FRWF)
D999 B7 1363 OR A
D99A 20 0B 1364 JR NZ,FMSSETS2;WRITE
D99C 21 68 E0 1365 LD HL,#FNAM
D99F 06 23 1366 LD B,35
D9A1 AF 1367 XOR A
D9A2 1368 DTBLCL
D9A2 77 1369 LD (HL),A
D9A3 23 1370 INC HL
D9A4 18 FC 1371 DJNZ DTBLCL
D9A6 C9 1372 RET
D9A7 1373
D9A7 1374 FMSSETS2;WRITE
D9A7 ED 5B 52 E0 1375 LD A,(#MORF)
D9AB 06 20 1376 LD B,32
D9AD AF 1377 RVFACL
D9AD CD 18 E0 1378 XOR A
D9AE 1379 CALL #LDDA
D9B1 13 1380 INC DE
D9B2 18 F9 1381 DJNZ RVFACL
D9B4 C9 1382 RET
D9B5 1383
D9B5 1384 ;/Bバイト分をエリアにSET/
D9B5 1385
D9B5 1386 JITS
D9B5 3A A1 E0 1387 LD A,(#FRWF)
D9B8 B7 1388 OR A
D9B9 20 0C 1389 JR NZ,JITS2;WRITE
D9BB 79 1390 JITSPLP
D9BC FE 20 1391 LD A,C
D9BC FE 20 1392 CP 32
D9BE CA 12 E0 1393 CALL NZ,#LDAHL
D9C1 12 1394 LD (DE),A
D9C2 23 1395 INC HL
D9C3 13 1396 INC DE
D9C4 18 F5 1397 DJNZ JITSPLP
D9C6 C9 1398 RET
D9C7 1399
D9C7 1400 JITS2;WRITE
D9C7 B7 1401 EX DE,HL
D9C8 C9 1402 NAGRLP
D9C8 79 1403 LD A,C
D9C9 FE 20 1404 CP 32
D9C9 FE 20 1405 LD A,(HL)
D9CC CA 18 E0 1406 CALL #LDDA
D9CF 23 1407 INC HL
D9D0 13 1408 INC DE
D9D1 18 F5 1409 DJNZ NAGRLP
D9D3 C9 1410 RET
D9D4 1411
D9D4 1412 ;/空回りをさがす/
D9D4 1413
D9D4 1414 DSECO;IN MORF OUT MORF
D9D4 CD 53 DD 1415 CALL SETFNDT;FORMAT割
D9D7 06 00 1416 LD B,0
D9D9 2A 52 E0 1417 LD HL,(#MORF)
D9DC 18 20 00 1418 LD DE,32
D9DD C9 1419 DSECLP
D9DF CD 12 E0 1420 CALL #LDAHL

```

▶PC-6001ユーザーだった頃、ナイコンの友人 Tel にばかにされ「ナイコンのぶんざい」と
 と思っていたらX68000を買った。だが、本日ACE-HDを買い、大声で言う。「HDをつけ
 てみろ！」これて私もX68000のユーザーだ。

城之内 一茂 (18) 千葉県

D9E2 DD BE 00 1421 CP (IX+0),END
D9E5 28 0B 1422 JP Z,DSRCRT
D9E7 DD BE 01 1423 CP (IX+1),NILL
D9EA 28 1C 1424 JR Z,DSRCRT
D9EC 19 1425 ADD HL,DE ;NEXT
D9ED 18 F0 1426 D9E2,DE:DE:CP ;


```

DCDE 3E 30      1870 LD A,'0' ;0を表示
DCDE CD 18 E0   1871 CALL #LDDEA ;
DCDE 13         1872 INC DE
DCDE 13         1873 RET
DCDE 13         1874
DCDE 13         1875 HEX2 ;IN A
DCDE F5         1876 PUSH AF
DCDE E6 F0      1877 AND #F0 ;4BIT分抽出
DCDE 0F         1878 RRCA
DCDE 0F         1879 RRCA
DCDE 0F         1880 RRCA
DCDE 0F         1881 RRCA
DCDE CD F1 DC   1882 CALL #HEXSB ;4BIT分
DCDE F1         1883 POP AF
DCDE E6 0F      1884 AND #F ;下位4BIT
DCDE 0F         1885 HEXSB
DCDE B7         1886 OR A
DCDE 28 01      1887 JR 2,HEXSB2
DCDE 0C         1888 INC C ;C=NOT 0 FLAG
DCDE 0C         1889 HEXSB2
DCDE C6 30      1890 ADD A,'0'
DCDE FE 3A      1891 CP '0'+1
DCDE 38 02      1892 JR C,HEXSK1 ;IF 数字
DCDE C6 07      1893 ADD A,'A'-'0'-1
DCDE 0C         1894 HEXSK1
DCDE CD 18 E0   1895 CALL #LDDEA
DD00 79         1896 LD A,C ;IF 0 なら
DD01 87         1897 OR A ;0じゃなくて
DD02 3E 20      1898 LD A,' ' ;' 'を出力
DD04 CD 18 E0   1899 CALL #LDDEA
DD07 13         1900 INC DE
DD08 C9         1901 RET
DD09           1902
DD09           1903 ;(DATA SET)
DD09           1904 ;
DD09           1905 TENS0
DD09 21 88 E0   1906 LD HL,#NAME ;NAME
DD0C 06 00      1907 CP '0'+1
DD0E CD 17 DD   1908 CALL BLDIR ;
DD11 13         1909 INC DE
DD12 06 03      1910 LD B,3 ;3文字
DD14 CD 17 DD   1911 JP BLDIR ;しまった！
DD17           1912 BLDIR
DD17 7E         1913 LD A,(HL)
DD18 23         1914 INC HL
DD19 CD 18 E0   1915 CALL #LDDEA
DD1C 13         1916 INC DE
DD1D 10 F8      1917 DJNZ BLDIR
DD1F C9         1918 RET
DD20           1919 ;DIR=0 ;32バイトCLEAR
DD20 D5         1920 PUSH DE
DD21 06 20      1921 LD B,32
DD22           1922 BLDIRLP
DD23 3E 20      1923 LD A,' '
DD25 CD 18 E0   1924 CALL #LDDEA
DD28 13         1925 INC DE
DD29 10 F8      1926 DJNZ BLDIRLP
DD2B D1         1927 POP DE
DD2C C9         1928 RET
DD2D           1929
DD2D           1930 ;(属性の表示用DATA SET)
DD2D           1931 ;
DD2D           1932 OPTZ
DD2D 3A 89 E0   1933 LD A,(#FDDP+2)
DD30 B7         1934 OR A
DD31 3E 50      1935 LD A,'P'
DD33 28 03      1936 JR 2,OPTZ
DD35 CD 18 E0   1937 CALL #LDDEA ;'P' SET
DD38           1938 OPTZ2
DD38 13         1939 INC DE
DD39 3A 8A E0   1940 LD A,(#FDDP+3)
DD3C B7         1941 OR A
DD3D 21 C2 DD   1942 LD HL,ASCDT
DD40 20 80      1943 JR NZ,OPTZ2 ;IF DIR
DD42 3A 87 E0   1944 LD A,(#FDDP)
DD45 06 04      1945 LD B,4
DD47           1946 OPTZLP ;Bin,Bas,Ascの判定
DD47 23         1947 INC HL
DD48 0F         1948 RRCA
DD49 38 02      1949 JR C,OPTZ2
DD4B 10 FA      1950 DJNZ OPTZLP
DD4D           1951 OPTZ3
DD4D 7E         1952 LD A,(HL) ;(ASCDT+?)
DD4E CD 18 E0   1953 CALL #LDDEA
DD51 13         1954 INC DE
DD52 C9         1955 RET
DD53           1956
DD53           1957 ;(FORMAT側DATA-INDEXのSET)
DD53           1958 ;
DD53           1959 SETFNDT
DD53 3A 97 E0   1960 LD A,(#MAC)
DD56 DD 21 3C E1 1961 LD IX,FNDT1 ;X12HD
DD5A 3D         1962 DEC A
DD5B C8         1963 RET 2
DD5C DD 21 44 E1 1964 LD IX,FNDT2 ;MS-DOS 2HD
DD60 3D         1965 DEC A
DD61 C8         1966 RET 2
DD62 DD 21 3C E1 1967 LD IX,FNDT1 ;X12D
DD66 3D         1968 DEC A
DD67 C8         1969 RET 2
DD68 DD 21 44 E1 1970 LD IX,FNDT2 ;MS-DOS 2D
DD6C C9         1971 RET
DD6D           1972
DD6D           1973 ;/ 256バイト転送 /
DD6D           1974 ;
DD6D           1975 TRS256 ;IN BF OUT BF
DD6D 06 00      1976 LD 0
DD6F 2A 91 E0   1977 LD HL,(#BF) ;(#BF)から
DD72 CD 79 DD   1978 CALL TRSB
DD75 22 91 E0   1979 LD (#BF),HL
DD78 C9         1980 RET
DD79           1981 TRSB
DD79 11 00 ER   1982 LD DE,P256 ;P256へ
DD7C           1983 TRSLP256
DD7C CD 12 E0   1984 CALL #LDAHL
DD7F 23         1985 INC HL
DD80 12         1986 LD (DE),A
DD81 13         1987 INC DE
DD82 10 F8      1988 DJNZ TRSLP256
DD84 C9         1989 RET
DD85           1990
DD85           1991 ;/ ドライブ番号の判定 /
DD85           1992 ;
DD85           1993 ;
DD85           1994 SETDN ;IN B,(DE)
DD85 78         1995 LD A,B
DD86 B7         1996 OR A
DD87 C8         1997 RET 2 ;IF DATA=0
DD88 13         1998 INC DE
DD89 1A         1999 LD A,(DE)
DD8A 1B         2000 DEC DE
DD8B FE 3A      2001 CP ' ' ;'X'?
DD8C DD 20 27   2002 JR NZ,SETDNRT2
DD8F 1A         2003 LD A,(DE)
DD90 FE 61      2004 CP 'h'
DD92 38 02      2005 JR C,SETDN
DD94 D6 20      2006 SUB 'a'-'A'
DD96           2007 SETDNRSK
DD96 D6 11      2008 SUB 'A'
DD98 FE 3A      2009 CP ' ' ;'A'+1
DD9A 38 07      2010 JR C,SETDNRT ;IF 正しい
DD9C 3E 04      2011 LD A,4
DD9E 32 8C E0   2012 LD (#STOPT),A
DDA1 37         2013 SCF ;エラー
DDA2 C9         2014 RET
DDA3           2015 SETDNRT
DDA3 32 8B E0   2016 LD (#DN),A ;SET
DDA6 21 C0 EC   2017 LD HL,#MAC4
DDA8 C9         2018 PUSH BC
DDAA CD 88 DD   2019 CALL DNST2
DDAD C1         2020 POP BC
DDAE 7E         2021 LD A,(HL)
DDAF CD 32 97 E0 2022 LD (#MAC),A ;FORMAT SET
DDB2 13         2023 INC DE ;
DDB3 13         2024 INC DE ;POINTERを
DDB4 05         2025 DEC B ;'X'の次へ
DDB5 05         2026 DEC B
DDB6           2027 SETDNRT2
DDB6 B7         2028 OR A
DDB7 C9         2029 RET
DDB8           2030
DDB8           2031 DNST2 ;HL=HL+(#DN)
DDBB C5         2032 PUSH BC
DDBB 06 00      2033 LD B,0
DDDB 7E         1952 LD A,(HL) ;(ASCDT+?)
DDDE CD 18 E0   1953 CALL #LDDEA
DD51 13         1954 INC DE
DD52 C9         1955 RET
DD53           1956
DD53           1957 ;(FORMAT側DATA-INDEXのSET)
DD53           1958 ;
DD53           1959 SETFNDT
DD53 3A 97 E0   1960 LD A,(#MAC)
DD56 DD 21 3C E1 1961 LD IX,FNDT1 ;X12HD
DD5A 3D         1962 DEC A
DD5B C8         1963 RET 2
DD5C DD 21 44 E1 1964 LD IX,FNDT2 ;MS-DOS 2HD
DD60 3D         1965 DEC A
DD61 C8         1966 RET 2
DD62 DD 21 3C E1 1967 LD IX,FNDT1 ;X12D
DD66 3D         1968 DEC A
DD67 C8         1969 RET 2
DD68 DD 21 44 E1 1970 LD IX,FNDT2 ;MS-DOS 2D
DD6C C9         1971 RET
DD6D           1972
DD6D           1973 ;/ 256バイト転送 /
DD6D           1974 ;
DD6D           1975 TRS256 ;IN BF OUT BF
DD6D 06 00      1976 LD 0
DD6F 2A 91 E0   1977 LD HL,(#BF) ;(#BF)から
DD72 CD 79 DD   1978 CALL TRSB
DD75 22 91 E0   1979 LD (#BF),HL
DD78 C9         1980 RET
DD79           1981 TRSB
DD79 11 00 ER   1982 LD DE,P256 ;P256へ
DD7C           1983 TRSLP256
DD7C CD 12 E0   1984 CALL #LDAHL
DD7F 23         1985 INC HL
DD80 12         1986 LD (DE),A
DD81 13         1987 INC DE
DD82 10 F8      1988 DJNZ TRSLP256
DD84 C9         1989 RET
DD85           1990
DD85           1991 ;/ ドライブ番号の判定 /
DD85           1992 ;
DD85           1993 ;
DD85           1994 SETDN ;IN B,(DE)
DD85 78         1995 LD A,B
DD86 B7         1996 OR A
DD87 C8         1997 RET 2 ;IF DATA=0
DD88 13         1998 INC DE
DD89 1A         1999 LD A,(DE)
DD8A 1B         2000 DEC DE
DD8B FE 3A      2001 CP ' ' ;'X'?
DD8C DD 20 27   2002 JR NZ,SETDNRT2
DD8F 1A         2003 LD A,(DE)
DD90 FE 61      2004 CP 'h'
DD92 38 02      2005 JR C,SETDN
DD94 D6 20      2006 SUB 'a'-'A'
DD96           2007 SETDNRSK
DD96 D6 11      2008 SUB 'A'
DD98 FE 3A      2009 CP ' ' ;'A'+1
DD9A 38 07      2010 JR C,SETDNRT ;IF 正しい
DD9C 3E 04      2011 LD A,4
DD9E 32 8C E0   2012 LD (#STOPT),A
DDA1 37         2013 SCF ;エラー
DDA2 C9         2014 RET
DDA3           2015 SETDNRT
DDA3 32 8B E0   2016 LD (#DN),A ;SET
DDA6 21 C0 EC   2017 LD HL,#MAC4
DDA8 C9         2018 PUSH BC
DDAA CD 88 DD   2019 CALL DNST2
DDAD C1         2020 POP BC
DDAE 7E         2021 LD A,(HL)
DDAF CD 32 97 E0 2022 LD (#MAC),A ;FORMAT SET
DDB2 13         2023 INC DE ;
DDB3 13         2024 INC DE ;POINTERを
DDB4 05         2025 DEC B ;'X'の次へ
DDB5 05         2026 DEC B
DDB6           2027 SETDNRT2
DDB6 B7         2028 OR A
DDB7 C9         2029 RET
DDB8           2030
DDB8           2031 DNST2 ;HL=HL+(#DN)
DDBB C5         2032 PUSH BC
DDBB 06 00      2033 LD B,0
DDDB 7E         1952 LD A,(HL) ;(ASCDT+?)
DDDE CD 18 E0   1953 CALL #LDDEA
DD51 13         1954 INC DE
DD52 C9         1955 RET
DD53           1956
DD53           1957 ;(FORMAT側DATA-INDEXのSET)
DD53           1958 ;
DD53           1959 SETFNDT
DD53 3A 97 E0   1960 LD A,(#MAC)
DD56 DD 21 3C E1 1961 LD IX,FNDT1 ;X12HD
DD5A 3D         1962 DEC A
DD5B C8         1963 RET 2
DD5C DD 21 44 E1 1964 LD IX,FNDT2 ;MS-DOS 2HD
DD60 3D         1965 DEC A
DD61 C8         1966 RET 2
DD62 DD 21 3C E1 1967 LD IX,FNDT1 ;X12D
DD66 3D         1968 DEC A
DD67 C8         1969 RET 2
DD68 DD 21 44 E1 1970 LD IX,FNDT2 ;MS-DOS 2D
DD6C C9         1971 RET
DD6D           1972
DD6D           1973 ;/ 256バイト転送 /
DD6D           1974 ;
DD6D           1975 TRS256 ;IN BF OUT BF
DD6D 06 00      1976 LD 0
DD6F 2A 91 E0   1977 LD HL,(#BF) ;(#BF)から
DD72 CD 79 DD   1978 CALL TRSB
DD75 22 91 E0   1979 LD (#BF),HL
DD78 C9         1980 RET
DD79           1981 TRSB
DD79 11 00 ER   1982 LD DE,P256 ;P256へ
DD7C           1983 TRSLP256
DD7C CD 12 E0   1984 CALL #LDAHL
DD7F 23         1985 INC HL
DD80 12         1986 LD (DE),A
DD81 13         1987 INC DE
DD82 10 F8      1988 DJNZ TRSLP256
DD84 C9         1989 RET
DD85           1990
DD85           1991 ;/ ドライブ番号の判定 /
DD85           1992 ;
DD85           1993 ;
DD85           1994 SETDN ;IN B,(DE)
DD85 78         1995 LD A,B
DD86 B7         1996 OR A
DD87 C8         1997 RET 2 ;IF DATA=0
DD88 13         1998 INC DE
DD89 1A         1999 LD A,(DE)
DD8A 1B         2000 DEC DE
DD8B FE 3A      2001 CP ' ' ;'X'?
DD8C DD 20 27   2002 JR NZ,SETDNRT2
DD8F 1A         2003 LD A,(DE)
DD90 FE 61      2004 CP 'h'
DD92 38 02      2005 JR C,SETDN
DD94 D6 20      2006 SUB 'a'-'A'
DD96           2007 SETDNRSK
DD96 D6 11      2008 SUB 'A'
DD98 FE 3A      2009 CP ' ' ;'A'+1
DD9A 38 07      2010 JR C,SETDNRT ;IF 正しい
DD9C 3E 04      2011 LD A,4
DD9E 32 8C E0   2012 LD (#STOPT),A
DDA1 37         2013 SCF ;エラー
DDA2 C9         2014 RET
DDA3           2015 SETDNRT
DDA3 32 8B E0   2016 LD (#DN),A ;SET
DDA6 21 C0 EC   2017 LD HL,#MAC4
DDA8 C9         2018 PUSH BC
DDAA CD 88 DD   2019 CALL DNST2
DDAD C1         2020 POP BC
DDAE 7E         2021 LD A,(HL)
DDAF CD 32 97 E0 2022 LD (#MAC),A ;FORMAT SET
DDB2 13         2023 INC DE ;
DDB3 13         2024 INC DE ;POINTERを
DDB4 05         2025 DEC B ;'X'の次へ
DDB5 05         2026 DEC B
DDB6           2027 SETDNRT2
DDB6 B7         2028 OR A
DDB7 C9         2029 RET
DDB8           2030
DDB8           2031 DNST2 ;HL=HL+(#DN)
DDBB C5         2032 PUSH BC
DDBB 06 00      2033 LD B,0
DDDB 7E         1952 LD A,(HL) ;(ASCDT+?)
DDDE CD 18 E0   1953 CALL #LDDEA
DD51 13         1954 INC DE
DD52 C9         1955 RET
DD53           1956
DD53           1957 ;(FORMAT側DATA-INDEXのSET)
DD53           1958 ;
DD53           1959 SETFNDT
DD53 3A 97 E0   1960 LD A,(#MAC)
DD56 DD 21 3C E1 1961 LD IX,FNDT1 ;X12HD
DD5A 3D         1962 DEC A
DD5B C8         1963 RET 2
DD5C DD 21 44 E1 1964 LD IX,FNDT2 ;MS-DOS 2HD
DD60 3D         1965 DEC A
DD61 C8         1966 RET 2
DD62 DD 21 3C E1 1967 LD IX,FNDT1 ;X12D
DD66 3D         1968 DEC A
DD67 C8         1969 RET 2
DD68 DD 21 44 E1 1970 LD IX,FNDT2 ;MS-DOS 2D
DD6C C9         1971 RET
DD6D           1972
DD6D           1973 ;/ 256バイト転送 /
DD6D           1974 ;
DD6D           1975 TRS256 ;IN BF OUT BF
DD6D 06 00      1976 LD 0
DD6F 2A 91 E0   1977 LD HL,(#BF) ;(#BF)から
DD72 CD 79 DD   1978 CALL TRSB
DD75 22 91 E0   1979 LD (#BF),HL
DD78 C9         1980 RET
DD79           1981 TRSB
DD79 11 00 ER   1982 LD DE,P256 ;P256へ
DD7C           1983 TRSLP256
DD7C CD 12 E0   1984 CALL #LDAHL
DD7F 23         1985 INC HL
DD80 12         1986 LD (DE),A
DD81 13         1987 INC DE
DD82 10 F8      1988 DJNZ TRSLP256
DD84 C9         1989 RET
DD85           1990
DD85           1991 ;/ ドライブ番号の判定 /
DD85           1992 ;
DD85           1993 ;
DD85           1994 SETDN ;IN B,(DE)
DD85 78         1995 LD A,B
DD86 B7         1996 OR A
DD87 C8         1997 RET 2 ;IF DATA=0
DD88 13         1998 INC DE
DD89 1A         1999 LD A,(DE)
DD8A 1B         2000 DEC DE
DD8B FE 3A      2001 CP ' ' ;'X'?
DD8C DD 20 27   2002 JR NZ,SETDNRT2
DD8F 1A         2003 LD A,(DE)
DD90 FE 61      2004 CP 'h'
DD92 38 02      2005 JR C,SETDN
DD94 D6 20      2006 SUB 'a'-'A'
DD96           2007 SETDNRSK
DD96 D6 11      2008 SUB 'A'
DD98 FE 3A      2009 CP ' ' ;'A'+1
DD9A 38 07      2010 JR C,SETDNRT ;IF 正しい
DD9C 3E 04      2011 LD A,4
DD9E 32 8C E0   2012 LD (#STOPT),A
DDA1 37         2013 SCF ;エラー
DDA2 C9         2014 RET
DDA3           2015 SETDNRT
DDA3 32 8B E0   2016 LD (#DN),A ;SET
DDA6 21 C0 EC   2017 LD HL,#MAC4
DDA8 C9         2018 PUSH BC
DDAA CD 88 DD   2019 CALL DNST2
DDAD C1         2020 POP BC
DDAE 7E         2021 LD A,(HL)
DDAF CD 32 97 E0 2022 LD (#MAC),A ;FORMAT SET
DDB2 13         2023 INC DE ;
DDB3 13         2024 INC DE ;POINTERを
DDB4 05         2025 DEC B ;'X'の次へ
DDB5 05         2026 DEC B
DDB6           2027 SETDNRT2
DDB6 B7         2028 OR A
DDB7 C9         2029 RET
DDB8           2030
DDB8           2031 DNST2 ;HL=HL+(#DN)
DDBB C5         2032 PUSH BC
DDBB 06 00      2033 LD B,0
DDDB 7E         1952 LD A,(HL) ;(ASCDT+?)
DDDE CD 18 E0   1953 CALL #LDDEA
DD51 13         1954 INC DE
DD52 C9         1955 RET
DD53           1956
DD53           1957 ;(FORMAT側DATA-INDEXのSET)
DD53           1958 ;
DD53           1959 SETFNDT
DD53 3A 97 E0   1960 LD A,(#MAC)
DD56 DD 21 3C E1 1961 LD IX,FNDT1 ;X12HD
DD5A 3D         1962 DEC A
DD5B C8         1963 RET 2
DD5C DD 21 44 E1 1964 LD IX,FNDT2 ;MS-DOS 2HD
DD60 3D         1965 DEC A
DD61 C8         1966 RET 2
DD62 DD 21 3C E1 1967 LD IX,FNDT1 ;X12D
DD66 3D         1968 DEC A
DD67 C8         1969 RET 2
DD68 DD 21 44 E1 1970 LD IX,FNDT2 ;MS-DOS 2D
DD6C C9         1971 RET
DD6D           1972
DD6D           1973 ;/ 256バイト転送 /
DD6D           1974 ;
DD6D           1975 TRS256 ;IN BF OUT BF
DD6D 06 00      1976 LD 0
DD6F 2A 91 E0   1977 LD HL,(#BF) ;(#BF)から
DD72 CD 79 DD   1978 CALL TRSB
DD75 22 91 E0   1979 LD (#BF),HL
DD78 C9         1980 RET
DD79           1981 TRSB
DD79 11 00 ER   1982 LD DE,P256 ;P256へ
DD7C           1983 TRSLP256
DD7C CD 12 E0   1984 CALL #LDAHL
DD7F 23         1985 INC HL
DD80 12         1986 LD (DE),A
DD81 13         1987 INC DE
DD82 10 F8      1988 DJNZ TRSLP256
DD84 C9         1989 RET
DD85           1990
DD85           1991 ;/ ドライブ番号の判定 /
DD85           1992 ;
DD85           1993 ;
DD85           1994 SETDN ;IN B,(DE)
DD85 78         1995 LD A,B
DD86 B7         1996 OR A
DD87 C8         1997 RET 2 ;IF DATA=0
DD88 13         1998 INC DE
DD89 1A         1999 LD A,(DE)
DD8A 1B         2000 DEC DE
DD8B FE 3A      2001 CP ' ' ;'X'?
DD8C DD 20 27   2002 JR NZ,SETDNRT2
DD8F 1A         2003 LD A,(DE)
DD90 FE 61      2004 CP 'h'
DD92 38 02      2005 JR C,SETDN
DD94 D6 20      2006 SUB 'a'-'A'
DD96           2007 SETDNRSK
DD96 D6 11      2008 SUB 'A'
DD98 FE 3A      2009 CP ' ' ;'A'+1
DD9A 38 07      2010 JR C,SETDNRT ;IF 正しい
DD9C 3E 04      2011 LD A,4
DD9E 32 8C E0   2012 LD (#STOPT),A
DDA1 37         2013 SCF ;エラー
DDA2 C9         2014 RET
DDA3           2015 SETDNRT
DDA3 32 8B E0   2016 LD (#DN),A ;SET
DDA6 21 C0 EC   2017 LD HL,#MAC4
DDA8 C9         2018 PUSH BC
DDAA CD 88 DD   2019 CALL DNST2
DDAD C1         2020 POP BC
DDAE 7E         2021 LD A,(HL)
DDAF CD 32 97 E0 2022 LD (#MAC),A ;FORMAT SET
DDB2 13         2023 INC DE ;
DDB3 13         2024 INC DE ;POINTERを
DDB4 05         2025 DEC B ;'X'の次へ
DDB5 05         2026 DEC B
DDB6           2027 SETDNRT2
DDB6 B7         2028 OR A
DDB7 C9         2029 RET
DDB8           2030
DDB8           2031 DNST2 ;HL=HL+(#DN)
DDBB C5         2032 PUSH BC
DDBB 06 00      2033 LD B,0
DDDB 7E         1952 LD A,(HL) ;(ASCDT+?)
DDDE CD 18 E0   1953 CALL #LDDEA
DD51 13         1954 INC DE
DD52 C9         1955 RET
DD53           1956
DD53           1957 ;(FORMAT側DATA-INDEXのSET)
DD53           1958 ;
DD53           1959 SETFNDT
DD53 3A 97 E0   1960 LD A,(#MAC)
DD56 DD 21 3C E1 1961 LD IX,FNDT1 ;X12HD
DD5A 3D         1962 DEC A
DD5B C8         1963 RET 2
DD5C DD 21 44 E1 1964 LD IX,FNDT2 ;MS-DOS 2HD
DD60 3D         1965 DEC A
DD61 C8         1966 RET 2
DD62 DD 21 3C E1 1967 LD IX,FNDT1 ;X12D
DD66 3D         1968 DEC A
DD67 C8         1969 RET 2
DD68 DD 21 44 E1 1970 LD IX,FNDT2 ;MS-DOS 2D
DD6C C9         1971 RET
DD6D           1972
DD6D           1973 ;/ 256バイト転送 /
DD6D           1974 ;
DD6D           1975 TRS256 ;IN BF OUT BF
DD6D 06 00      1976 LD 0
DD6F 2A 91 E0   1977 LD HL,(#BF) ;(#BF)から
DD72 CD 79 DD   1978 CALL TRSB
DD75 22 91 E0   1979 LD (#BF),HL
DD78 C9         1980 RET
DD79           1981 TRSB
DD79 11 00 ER   1982 LD DE,P256 ;P256へ
DD7C           1983 TRSLP256
DD7C CD 12 E0   1984 CALL #LDAHL
DD7F 23         1985 INC HL
DD80 12         1986 LD (DE),A
DD81 13         1987 INC DE
DD82 10 F8      1988 DJNZ TRSLP256
DD84 C9         1989 RET
DD85           1990
DD85           1991 ;/ ドライブ番号の判定 /
DD85           1992 ;
DD85           1993 ;
DD85           1994 SETDN ;IN B,(DE)
DD85 78         1995 LD A,B
DD86 B7         1996 OR A
DD87 C8         1997 RET 2 ;IF DATA=0
DD88 13         1998 INC DE
DD89 1A         1999 LD A,(DE)
DD8A 1B         2000 DEC DE
DD8B FE 3A      2001 CP ' ' ;'X'?
DD8C DD 20 27   2002 JR NZ,SETDNRT2
DD8F 1A         2003 LD A,(DE)
DD90 FE 61      2004 CP 'h'
DD92 38 02      2005 JR C,SETDN
DD94 D6 20      2006 SUB 'a'-'A'
DD96           2007 SETDNRSK
DD96 D6 11      2008 SUB 'A'
DD98 FE 3A      2009 CP ' ' ;'A'+1
DD9A 38 07      2010 JR C,SETDNRT ;IF 正しい
DD9C 3E 04      2011 LD A,4
DD9E 32 8C E0   2012 LD (#STOPT),A
DDA1 37         2013 SCF ;エラー
DDA2 C9         2014 RET
DDA3           2015 SETDNRT
DDA3 32 8B E0   2016 LD (#DN),A ;SET
DDA6 21 C0 EC   2017 LD HL,#MAC4
DDA8 C9         2018 PUSH BC
DDAA CD 88 DD   2019 CALL DNST2
DDAD C1         2020 POP BC
DDAE 7E         2021 LD A,(HL)
DDAF CD 32 97 E0 2022 LD (#MAC),A ;FORMAT SET
DDB2 13         2023 INC DE ;
DDB3 13         2024 INC DE ;POINTERを
DDB4 05         2025 DEC B ;'X'の次へ
DDB5 05         2026 DEC B
DDB6           2027 SETDNRT2
DDB6 B7         2028 OR A
DDB7 C9         2029 RET
DDB8           2030
DDB8           2031 DNST2 ;HL=HL+(#DN)
DDBB C5         2032 PUSH BC
DDBB 06 00      2033 LD B,0
DDDB 7E         1952 LD A,(HL) ;(ASCDT+?)
DDDE CD 18 E0   1953 CALL #LDDEA
DD51 13         1954 INC DE
DD52 C9         1955 RET
DD53           1956
DD53           1957 ;(FORMAT側DATA-INDEXのSET)
DD53           1958 ;
DD53           1959 SETFNDT
DD53 3A 97 E0   1960 LD A,(#MAC)
DD56 DD 21 3C E1 1961 LD IX,FNDT1 ;X12HD
DD5A 3D         1962 DEC A
DD5B C8         1963 RET 2
DD5C DD 21 44 E1 1964 LD IX,FNDT2 ;MS-DOS 2HD
DD60 3D         1965 DEC A
DD61 C8         1966 RET 2
DD62 DD 21 3C E1 1967 LD IX,FNDT1 ;X12D
DD66 3D         1968 DEC A
DD67 C8         1969 RET 2
DD68 DD 21 44 E1 1970 LD IX,FNDT2 ;MS-DOS 2D
DD6C C9         1971 RET
DD6D           1972
DD6D           1973 ;/ 256バイト転送 /
DD6D           1974 ;
DD6D           1975 TRS256 ;IN BF OUT BF
DD6D 06 00      1976 LD 0
DD6F 2A 91 E0   1977 LD HL,(#BF) ;(#BF)から
DD72 CD 79 DD   1978 CALL TRSB
DD75 22 91 E0   1979 LD (#BF),HL
DD78 C9         1980 RET
DD79           1981 TRSB
DD79 11 00 ER   1982 LD DE,P256 ;P256へ
DD7C           1983 TRSLP256
DD7C CD 12 E0   1984 CALL #LDAHL
DD7F 23         1985 INC HL
DD80 12         1986 LD (DE),A
DD81 13         1987 INC DE
DD82 10 F8      1988 DJNZ TRSLP256
DD84 C9         1989 RET
DD85           1990
DD85           1991 ;/ ドライブ番号の判定 /
DD85           1992 ;
DD85           1993 ;
DD85           1994 SETDN ;IN B,(DE)
DD85 78         1995 LD A,B
DD86 B7         1996 OR A
DD87 C8         1997 RET 2 ;IF DATA=0
DD88 13         1998 INC DE
DD89 1A         1999 LD A,(DE)
DD8A 1B         2000 DEC DE
DD8B FE 3A      2001 CP ' ' ;'X'?
DD8C DD 20 27   2002 JR NZ,SETDNRT2
DD8F 1A         2003 LD A,(DE)
DD90 FE 61      2004 CP 'h'
DD92 38 02      2005 JR C,SETDN
DD94 D6 20      2006 SUB 'a'-'A'
DD96           2007 SETDNRSK
DD96 D6 11      2008 SUB 'A'
DD98 FE 3A      2009 CP ' ' ;'A'+1
DD9A 38 07      2010 JR C,SETDNRT ;IF 正しい
DD9C 3E 04      2011 LD A,4
DD9E 32 8C E0   2012 LD (#STOPT),A
DDA1 37         2013 SCF ;エラー
DDA2 C9         2014 RET
DDA3           2015 SETDNRT
DDA3 32 8B E0   2016 LD (#DN),A ;SET
DDA6 21 C0 EC   2017 LD HL,#MAC4
DDA8 C9         2018 PUSH BC
DDAA CD 88 DD   2019 CALL DNST2
DDAD C1         2020 POP BC
DDAE 7E         2021 LD A,(HL)
DDAF CD 32 97 E0 2022 LD (#MAC),A ;FORMAT SET
DDB2 13         2023 INC DE ;
DDB3 13         2024 INC DE ;POINTERを
DDB4 05         2025 DEC B ;'X'の次へ
DDB5 05         2026 DEC B
DDB6           2027 SETDNRT2
DDB6 B7         2028 OR A
DDB7 C9         2029 RET
DDB8           2030
DDB8           2031 DNST2 ;HL=HL+(#DN)
DDBB C5         2032 PUSH BC
DDBB 06 00      2033 LD B,0
DDDB 7E         1952 LD A,(HL) ;(ASCDT+?)
DDDE CD 18 E0   1953 CALL #LDDEA
DD51 13         1954 INC DE
DD52 C9         1955 RET
DD53           1956
DD53           1957 ;(FORMAT側DATA-INDEXのSET)
DD53           1958 ;
DD53           1959 SETFNDT
DD53 3A 97 E0   1960 LD A,(#MAC)
DD56 DD 21 3C E1 1961 LD IX,FNDT1 ;X12HD
DD5A 3D         1962 DEC A
DD5B C8         1963 RET 2
DD5C DD 21 44 E1 1964 LD IX,FNDT2 ;MS-DOS 2HD
DD60 3D         1965 DEC A
DD61 C8         1966 RET 2
DD62 DD 21 3C E1 1967 LD IX,FNDT1 ;X12D
DD66 3D         1968 DEC A
DD67 C8         1969 RET 2
DD68 DD 21 44 E1 1970 LD IX,FNDT2 ;MS-DOS 2D
DD6C C9         1971 RET
DD6D           1972
DD6D           1973 ;/ 256バイト転送 /
DD6D           1974 ;
DD6D           1975 TRS256 ;IN BF OUT BF
DD6D 06 00      1976 LD 0
DD6F 2A 91 E0   1977 LD HL,(#BF) ;(#BF)から
DD72 CD 79 DD   1978 CALL TRSB
DD75 22 91 E0   1979 LD (#BF),HL
DD78 C9         1980 RET
DD79           1981 TRSB
DD79 11 00 ER   1982 LD DE,P256 ;P256へ
DD7C           1983 TRSLP256
DD7C CD 12 E0   1984 CALL #LDAHL
DD7F 23         1985 INC HL

```

148 Oh! X 1990.7.

松永 直哉 (21) 東京都


```

E4AE 3A 97 E0 681 LD A, (#MAC)
E4B1 FE 01 682 CP
E4B3 20 05 603 JR NZ,SECMN1; IF X12HD以外
E4B5 7D 604 LD A,L; HLが#80
E4B6 FE 80 605 CP #80
E4B8 30 F3 606 JR NC,SECLP0; なるまで
E4BA 607 SECMN1
E4BA C5 608 PUSH BC
E4BC 22 95 E0 610 LD (#CRS),HL
E4BD B7 611 OR A
E4CE ED 42 612 SBC HL,BC; FATの最後か?
E4D0 30 12 613 JR NC,SECSK2; IF 最後
E4C4 CD AE E3 614 CALL SECFA2T2; 次は
E4C7 E1 615 POP HL; 次は
E4C9 C1 616 POP BC
E4C9 ED 5B 95 E0 617 LD DE, (#CRS); DE=FAT
E4CD 7A 618 LD A,D
E4CE B3 619 OR E
E4CF 20 DC 620 JR NZ,SECLP0; IF DE<0
E4D1 22 95 E0 621 LD (#CRS),HL; HL=未使用
E4D4 B7 622 OR A; クラスタ
E4D5 C9 623 RET
E4D6 624 SECSK2; ディスクが一杯です
E4D6 E1 625 POP HL
E4D7 C1 626 POP BC
E4D8 C3 3C E0 627 JP #ERR5; エラー
E4DB 628
E4DB 629 ; クラスタ番号を
E4DB 630 ; FATへ書き込む
E4DB 631 WFA2T2; IN CRS,CRS,FF,MAC
E4DB 3A 97 E0 632 LD A, (#MAC)
E4DB 3D 633 DEC A
E4DF 2B 09 634 JR 2,SPAT11
E4E1 3D 635 DEC A
E4E2 2B 21 636 JR 2,SPAT22
E4E4 3D 637 DEC A
E4E5 CA 52 E5 638 JP Z,SPAT33
E4E8 1B 18 639 JR SPAT22
E4EA 640
E4EA 641 SPAT11; X12HD
E4EA 2A 8D E0 642 LD HL, (#FF)
E4ED 5B 9B E0 643 LD DE, (CRS)
E4F1 19 644 ADD HL,DE
E4F2 EB 645 EX DE,HL; DE=FAT
E4F3 3A 95 E0 646 LD A, (#CRS)
E4F6 CD 96 E6 647 CALL LDDEA
E4F9 EB 648 EX DE,HL;
E4FA 11 80 00 649 LD DE, 128;
E4FD 19 650 ADD HL,DE
E4FE EB 651 EX DE,HL; DE=DE+128
E4FF 3A 96 E0 652 LD A, (#CRS+1)
E502 C3 96 E6 653 JP LDDEA
E505 654
E505 655 SPAT22; MS-DOS 2HD2D
E505 2A 90 EB 656 LD HL, (CRS)
E508 CB 3C 657 SRL H;
E50A CB 1D 658 RRL; HL=HL/2
E50C 00 659 EX AF,AF; C FLUG
E50D E5 660 PUSH HL
E50E 29 661 ADD HL,HL;
E50F D1 662 POP DE;
E510 19 663 ADD HL,DE; HL=(CRS)/2+3
E511 EB 664 EX DE,HL;
E512 2A 8D E0 665 LD HL, (#FF)
E515 19 666 ADD HL,DE;
E516 EB 667 EX DE,HL; HL=FATADR+
E517 00 668 EX AF,AF;
E518 3B 14 669 JR C,SPFN33
E51A 3A 95 E0 670 LD A, (#CRS)
E51D CD 96 E6 671 CALL LDDEA
E520 13 672 INC DE
E521 CD 8A E6 673 CALL LDDEA; もう1つの
E524 E6 F0 674 AND #F0; FATと合わせ
E526 47 675 LD B,A; 枝
E527 3A 96 E0 676 LD A, (#CRS+1)
E52A D0 677 OR B
E52B C3 96 E6 678 JP LDDEA
E52E 679 SPFN33
E52E 3A 96 E0 680 LD A, (#CRS+1)
E531 47 681 LD B,A
E532 3A 95 E0 682 LD A, (#CRS)
E535 B7 683 OR A
E536 17 684 RLA
E537 CB 10 685 RL B
E539 17 686 RLA
E53A CB 10 687 RL B
E53C 17 688 RLA; 以上
E53D CB 10 689 RL B; A8000001111111111を
E53F 17 690 RLA; A8111111111110000に
E540 CB 10 691 RL B; なおしている
E542 4F 692 LD C,A
E543 13 693 INC DE
E544 CD 8A E6 694 CALL LDDEA
E547 E6 0F 695 AND #0F
E549 B1 696 OR C; 合わせ枝
E54A CD 96 E6 697 CALL LDDEA
E54D 13 698 INC DE
E54E 78 699 LD A,B
E54F C3 96 E6 700 JP LDDEA
E552 701
E552 702 SPAT33; X1 2D
E552 2A 8D E0 703 LD HL, (#FF)
E555 ED 5B 9B E0 704 LD DE, (CRS)
E559 19 705 ADD HL,DE
E55A EB 706 EX DE,HL;
E55B 3A 95 E0 707 LD A, (#CRS)
E55E C3 96 E6 708 JP LDDEA
E561 709
E561 710 ; / DELETE FAT /
E561 711 ;
E561 712 DELPAT; IN ECRS
E561 713 LD HL, (#ECS)
E564 714 DFATLP
E564 22 95 E0 715 LD (#CRS),HL
E567 22 9B E0 716 LD (CRS),HL
E56A CD AE E3 717 CALL SECFA2T2; 位置
E56D 2A 95 E0 718 LD HL, (#CRS)
E570 E5 719 PUSH HL
E571 21 00 00 720 LD HL, 0; 0を書き込む
E574 22 95 E0 721 LD (#CRS),HL
E577 CD DB E4 722 CALL WFA2T2; WRITE
E57A E1 723 POP HL
E57B 3A 8F E0 724 LD A, (EDD); FATが
E57E B7 725 OR A; HLが、とされるまで
E57F 2B E3 726 JR 2,DFATLP
E581 C9 727 RET
E582 728
E582 729 ; ( FORMAT DATA SET )
E582 730 ;
E582 731 VAR; IN MAC
E582 C5 732 PUSH BC
E583 D5 733 PUSH DE
E584 E5 734 PUSH HL
E585 D5 735 PUSH IX
E587 3A 97 E0 736 LD A, (#MAC)
E58A DD 21 00 E1 737 LD IX, DVCDT1; X1 2HD
E58E 19 738 DEC IX
E58F 2B 12 739 JR 2,VALUEJP
E591 DD 21 0D E1 740 LD IX, DVCDT2; MS-DOS2HD
E595 D0 741 DEC A
E596 2B 08 742 JR 2,VALUEJP
E598 DD 21 1A E1 743 LD IX, DVCDT3; X1 2D
E59C 3D 744 DEC A
E59D 2B 04 745 JR 2,VALUEJP
E59F DD 21 27 E1 746 LD IX, DVCDT4; MS-DOS2D
E5A3 747 VALUEJP
E5A3 DD 6E 00 748
E5A6 DD 6E 01 749 LD L, (IX+0); セクター
E5A9 22 9F E0 750 LD H, (IX+1); の長さ
E5AC DD 7E 02 751 LD A, (IX+2); 1TRACK内の
E5AF 32 82 EB 752 LD (LSCT),A; セクター数
E5B2 DD 7E 03 753 LD A, (IX+3); 1 クラスタ内の
E5B5 32 95 EB 754 LD (RCL),A; クラスタの
E5B8 DD 5E 04 755 LD E, (IX+4);
E5BB DD 56 05 756 LD D, (IX+5); 大きさ
E5BE DD 53 93 EB 757 LD (CSBYT),DE; (バイト)
E5C2 2A 8F E0 758 LD HL, (ABSIZ); BUFFの大きさ
E5C5 CD 44 E6 759 CALL DIVND; #BSIZ/CSBYT
E5C8 32 8A EB 760 LD (BSCS),A; クラスタ間隔
E5CB DD 6E 06 761 LD L, (IX+6);
E5CE DD 6E 07 762 LD H, (IX+7); FAT開始位置
E5D1 22 83 EB 763 LD (RS),HL; レコード
E5D4 DD 6E 08 764 LD L, (IX+8); FAT終了位置
E5D7 DD 66 09 765 LD H, (IX+9);
E5DA DD 2E 05 EB 766 LD (RE),HL;
E5DD DD 7E 0A 767 LD A, (IX+10);
E5E0 32 87 EB 768 LD (LTR),A; 最終トラック
E5E3 DD 6E 08 769 LD L, (IX+11);
E5E6 DD 6E 0C 770 LD H, (IX+12);
E5E9 DD 2E 08 EB 771 LD (RRC),HL; 最終レコード
E5EC DD E1 772 POP IX
E5EE E1 773 POP HL
E5EF D1 774 POP DE
E5F0 C1 775 POP BC
E5F1 C9 776 RET
E5F2 777
E5F2 778 ; ( FATをDISKへ書き込む )
E5F2 779 ;
E5F2 780 FATW; IN FF,SLG,RS,RE
E5F2 781 LD A, 1;
E5F4 32 A1 E0 782 LD (#FRWF),A; WRITE
E5F7 CD 21 E6 783 CALL FATW;
E5FA 3A 97 E0 784 LD A, (#MAC)
E5FD FE 01 785 CP 1
E5FF 2B 18 786 JR Z,FATWTN2
E601 FE 03 787 CP 3
E603 2B 17 788 JR Z,FATWTN2
E605 21 03 00 789 LD HL, 3; MS-DOSはFATを
E608 22 83 EB 790 LD (RS),HL
E60B 23 791 INC HL; 2つ作る
E60C 22 85 EB 792 LD (RE),HL
E60F CD 21 E6 793 CALL FATW;
E612 21 01 00 794 LD HL, 1;
E615 22 83 EB 795 LD (RS),HL
E618 23 796 INC HL
E619 22 85 EB 797 LD (RE),HL
E61C 798 FATWTN2;
E61C AF 799 XOR A
E61D 32 A1 E0 800 LD (#FRWF),A
E620 C9 801 RET
E621 802
E621 803 FATRW; FAT読み書き
E621 D5 804 PUSH DE
E622 E5 805 PUSH HL
E623 2A 8D E0 806 LD HL, (#FF); FATADR
E626 22 91 E0 807 LD (#FF); BUFF
E629 2A 83 EB 808 LD HL, (RS); 開始REC
E62C 22 93 E0 809 LD (#REC),HL
E62F 810 FATRDL;
E62F CD 00 E0 811 CALL #WRREC
E632 3B 0D 812 JR C,PLTFT
E634 2A 85 EB 813 LD HL, (RE)
E637 ED 5B 93 E0 814 LD DE, (#REC)
E63B B7 815 OR A
E63C ED 52 816 SBC HL,DE; 最終?
E63E 30 EF 817 JR NC,FATRDL
E640 B7 818 OR A
E641 819 PLTFT
E641 E1 820 POP HL
E642 D1 821 POP DE
E643 C9 822 RET
E644 823
E644 824 ; たちの数値掛け算
E644 825 DIVD; DE=HL/DE,HL=HL MOD DE
E644 3E 10 826 LD A, 16
E646 4B 827 LD C,E
E647 42 828 LD B,D
E648 D0 829 EX DE,HL
E649 21 00 00 830 LD HL, 0
E64C EB 831 DIVLP1
E64C EB 832 EX DE,HL
E64D 29 833 ADD HL,HL
E64E EB 834 EX DE,HL
E64F ED 6A 835 ADC HL,HL
E651 CA 5D E6 836 JP Z,DIVLP4
E654 ED 42 837 SBC HL,BC
E656 30 04 838 JR NC,DIVLP3
E658 00 839 ADD HL,BC
E659 C3 5D E6 840 JP DIVLP4
E65C 841 DIVLP3
E65C 13 842 INC DE
E65D 843 DIVLP4
E65D 3D 844 DEC A
E65E C2 4C E6 845 JP NZ,DIVLP1
E661 7B 846 LD A,E
E662 C9 847 RET
E663 848
E663 849 ; たちの数値掛け算
E663 850 MLTHD; HL=HL*DE
E663 C5 851 PUSH BC
E664 D5 852 PUSH DE
E665 3E 10 853 LD A, 16
E667 44 854 LD B,H
E668 AD 855 LD C,L
E669 21 00 00 856 LD HL, 0
E66C 857 MLTLP
E66C 29 858 ADD HL,HL
E66D EB 859 EX DE,HL
E66E 29 860 ADD HL,HL
E66F EB 861 EX DE,HL
E670 D2 74 E6 862 JP NC,MULT2
E673 69 863 ADD HL,BC
E674 3D 864 MULT2
E675 30 F5 865 DEC A
E676 3D 866 JR NZ,MULTLP
E677 D1 867 POP DE
E678 C1 868 POP BC
E679 C9 869 RET
E67A 870
E67A 871 ; 以下はBUFFにアクセス
E67A 872 ; するための命令群
E67A 873 ;
E67A 874 ; バンクやI/Oに關係なく
E67A 875 ; BUFFを設定できる
E67A 876
E67A 877 LDHL; LD A, (HL)
E67A CD B3 E6 878 CALL BG10MM; バンク替えなど
E67D 78 879 LD A, (HL)
E67E D2 D9 E6 880 JP NC,BORST; もともとです
E681 C5 881 PUSH BC
E682 44 882 LD B,H
E683 AD 883 LD C,L
E684 ED 78 884 IN A, (C)
E686 C1 885 POP BC
E687 C3 D9 E6 886 JP BORST
E68A 887
E68A 888 LDDEA; LD A, (DE)
E68A EB 889 EX DE,HL
E68B CD 7A E6 890 CALL LDHL
E68E EB 891 EX DE,HL
E68F C9 892 RET
E690 893
E690 894 LDHL; LD (HL),A
E690 EB 895 EX DE,HL
E691 CD 96 E6 896 LD A, (DE)
E691 EB 897 EX DE,HL
E691 CD 96 E6 898 LD A, (DE)
E691 EB 899 EX DE,HL
E691 CD 96 E6 900 LD A, (DE)
E691 EB 901 EX DE,HL
E691 CD 96 E6 902 LD A, (DE)
E691 EB 903 EX DE,HL
E691 CD 96 E6 904 LD A, (DE)
E691 EB 905 EX DE,HL
E691 CD 96 E6 906 LD A, (DE)
E691 EB 907 EX DE,HL
E691 CD 96 E6 908 LD A, (DE)
E691 EB 909 EX DE,HL
E691 CD 96 E6 910 LD A, (DE)
E691 EB 911 EX DE,HL
E691 CD 96 E6 912 LD A, (DE)
E691 EB 913 EX DE,HL
E691 CD 96 E6 914 LD A, (DE)
E691 EB 915 EX DE,HL
E691 CD 96 E6 916 LD A, (DE)
E691 EB 917 EX DE,HL
E691 CD 96 E6 918 LD A, (DE)
E691 EB 919 EX DE,HL
E691 CD 96 E6 920 LD A, (DE)
E691 EB 921 EX DE,HL
E691 CD 96 E6 922 LD A, (DE)
E691 EB 923 EX DE,HL
E691 CD 96 E6 924 LD A, (DE)
E691 EB 925 EX DE,HL
E691 CD 96 E6 926 LD A, (DE)
E691 EB 927 EX DE,HL
E691 CD 96 E6 928 LD A, (DE)
E691 EB 929 EX DE,HL
E691 CD 96 E6 930 LD A, (DE)
E691 EB 931 EX DE,HL
E691 CD 96 E6 932 LD A, (DE)
E691 EB 933 EX DE,HL
E691 CD 96 E6 934 LD A, (DE)
E691 EB 935 EX DE,HL
E691 CD 96 E6 936 LD A, (DE)
E691 EB 937 EX DE,HL
E691 CD 96 E6 938 LD A, (DE)
E691 EB 939 EX DE,HL
E691 CD 96 E6 940 LD A, (DE)
E691 EB 941 EX DE,HL
E691 CD 96 E6 942 LD A, (DE)
E691 EB 943 EX DE,HL
E691 CD 96 E6 944 LD A, (DE)
E691 EB 945 EX DE,HL
E691 CD 96 E6 946 LD A, (DE)
E691 EB 947 EX DE,HL
E691 CD 96 E6 948 LD A, (DE)
E691 EB 949 EX DE,HL
E691 CD 96 E6 950 LD A, (DE)
E691 EB 951 EX DE,HL
E691 CD 96 E6 952 LD A, (DE)
E691 EB 953 EX DE,HL
E691 CD 96 E6 954 LD A, (DE)
E691 EB 955 EX DE,HL
E691 CD 96 E6 956 LD A, (DE)
E691 EB 957 EX DE,HL
E691 CD 96 E6 958 LD A, (DE)
E691 EB 959 EX DE,HL
E691 CD 96 E6 960 LD A, (DE)
E691 EB 961 EX DE,HL
E691 CD 96 E6 962 LD A, (DE)
E691 EB 963 EX DE,HL
E691 CD 96 E6 964 LD A, (DE)
E691 EB 965 EX DE,HL
E691 CD 96 E6 966 LD A, (DE)
E691 EB 967 EX DE,HL
E691 CD 96 E6 968 LD A, (DE)
E691 EB 969 EX DE,HL
E691 CD 96 E6 970 LD A, (DE)
E691 EB 971 EX DE,HL
E691 CD 96 E6 972 LD A, (DE)
E691 EB 973 EX DE,HL
E691 CD 96 E6 974 LD A, (DE)
E691 EB 975 EX DE,HL
E691 CD 96 E6 976 LD A, (DE)
E691 EB 977 EX DE,HL
E691 CD 96 E6 978 LD A, (DE)
E691 EB 979 EX DE,HL
E691 CD 96 E6 980 LD A, (DE)
E691 EB 981 EX DE,HL
E691 CD 96 E6 982 LD A, (DE)
E691 EB 983 EX DE,HL
E691 CD 96 E6 984 LD A, (DE)
E691 EB 985 EX DE,HL
E691 CD 96 E6 986 LD A, (DE)
E691 EB 987 EX DE,HL
E691 CD 96 E6 988 LD A, (DE)
E691 EB 989 EX DE,HL
E691 CD 96 E6 990 LD A, (DE)
E691 EB 991 EX DE,HL
E691 CD 96 E6 992 LD A, (DE)
E691 EB 993 EX DE,HL
E691 CD 96 E6 994 LD A, (DE)
E691 EB 995 EX DE,HL
E691 CD 96 E6 996 LD A, (DE)
E691 EB 997 EX DE,HL
E691 CD 96 E6 998 LD A, (DE)
E691 EB 999 EX DE,HL
E691 CD 96 E6 1000 LD A, (DE)
E691 EB 1001 EX DE,HL
E691 CD 96 E6 1002 LD A, (DE)
E691 EB 1003 EX DE,HL
E691 CD 96 E6 1004 LD A, (DE)
E691 EB 1005 EX DE,HL
E691 CD 96 E6 1006 LD A, (DE)
E691 EB 1007 EX DE,HL
E691 CD 96 E6 1008 LD A, (DE)
E691 EB 1009 EX DE,HL
E691 CD 96 E6 1010 LD A, (DE)
E691 EB 1011 EX DE,HL
E691 CD 96 E6 1012 LD A, (DE)
E691 EB 1013 EX DE,HL
E691 CD 96 E6 1014 LD A, (DE)
E691 EB 1015 EX DE,HL
E691 CD 96 E6 1016 LD A, (DE)
E691 EB 1017 EX DE,HL
E691 CD 96 E6 1018 LD A, (DE)
E691 EB 1019 EX DE,HL
E691 CD 96 E6 1020 LD A, (DE)
E691 EB 1021 EX DE,HL
E691 CD 96 E6 1022 LD A, (DE)
E691 EB 1023 EX DE,HL
E691 CD 96 E6 1024 LD A, (DE)
E691 EB 1025 EX DE,HL
E691 CD 96 E6 1026 LD A, (DE)
E691 EB 1027 EX DE,HL
E691 CD 96 E6 1028 LD A, (DE)
E691 EB 1029 EX DE,HL
E691 CD 96 E6 1030 LD A, (DE)
E691 EB 1031 EX DE,HL
E691 CD 96 E6 1032 LD A, (DE)
E691 EB 1033 EX DE,HL
E691 CD 96 E6 1034 LD A, (DE)
E691 EB 1035 EX DE,HL
E691 CD 96 E6 1036 LD A, (DE)
E691 EB 1037 EX DE,HL
E691 CD 96 E6 1038 LD A, (DE)
E691 EB 1039 EX DE,HL
E691 CD 96 E6 1040 LD A, (DE)
E691 EB 1041 EX DE,HL
E691 CD 96 E6 1042 LD A, (DE)
E691 EB 1043 EX DE,HL
E691 CD 96 E6 1044 LD A, (DE)
E691 EB 1045 EX DE,HL
E691 CD 96 E6 1046 LD A, (DE)
E691 EB 1047 EX DE,HL
E691 CD 96 E6 1048 LD A, (DE)
E691 EB 1049 EX DE,HL
E691 CD 96 E6 1050 LD A, (DE)
E691 EB 1051 EX DE,HL
E691 CD 96 E6 1052 LD A, (DE)
E691 EB 1053 EX DE,HL
E691 CD 96 E6 1054 LD A, (DE)
E691 EB 1055 EX DE,HL
E691 CD 96 E6 1056 LD A, (DE)
E691 EB 1057 EX DE,HL
E691 CD 96 E6 1058 LD A, (DE)
E691 EB 1059 EX DE,HL
E691 CD 96 E6 1060 LD A, (DE)
E691 EB 1061 EX DE,HL
E691 CD 96 E6 1062 LD A, (DE)
E691 EB 1063 EX DE,HL
E691 CD 96 E6 1064 LD A, (DE)
E691 EB 1065 EX DE,HL
E691 CD 96 E6 1066 LD A, (DE)
E691 EB 1067 EX DE,HL
E691 CD 96 E6 1068 LD A, (DE)
E691 EB 1069 EX DE,HL
E691 CD 96 E6 1070 LD A, (DE)
E691 EB 1071 EX DE,HL
E691 CD 96 E6 1072 LD A, (DE)
E691 EB 1073 EX DE,HL
E691 CD 96 E6 1074 LD A, (DE)
E691 EB 1075 EX DE,HL
E691 CD 96 E6 1076 LD A, (DE)
E691 EB 1077 EX DE,HL
E691 CD 96 E6 1078 LD A, (DE)
E691 EB 1079 EX DE,HL
E691 CD 96 E6 1080 LD A, (DE)
E691 EB 1081 EX DE,HL
E691 CD 96 E6 1082 LD A, (DE)
E691 EB 1083 EX DE,HL
E691 CD 96 E6 1084 LD A, (DE)
E691 EB 1085 EX DE,HL
E691 CD 96 E6 1086 LD A, (DE)
E691 EB 1087 EX DE,HL
E691 CD 96 E6 1088 LD A, (DE)
E691 EB 1089 EX DE,HL
E691 CD 96 E6 1090 LD A, (DE)
E691 EB 1091 EX DE,HL
E691 CD 96 E6 1092 LD A, (DE)
E691 EB 1093 EX DE,HL
E691 CD 96 E6 1094 LD A, (DE)
E691 EB 1095 EX DE,HL
E691 CD 96 E6 1096 LD A, (DE)
E691 EB 1097 EX DE,HL
E691 CD 96 E6 1098 LD A, (DE)
E691 EB 1099 EX DE,HL
E691 CD 96 E6 1100 LD A, (DE)
E691 EB 1101 EX DE,HL
E691 CD 96 E6 1102 LD A, (DE)
E691 EB 1103 EX DE,HL
E691 CD 96 E6 1104 LD A, (DE)
E691 EB 1105 EX DE,HL
E691 CD 96 E6 1106 LD A, (DE)
E691 EB 1107 EX DE,HL
E691 CD 96 E6 1108 LD A, (DE)
E691 EB 1109 EX DE,HL
E691 CD 96 E6 1110 LD A, (DE)
E691 EB 1111 EX DE,HL
E691 CD 96 E6 1112 LD A, (DE)
E691 EB 1113 EX DE,HL
E691 CD 96 E6 1114 LD A, (DE)
E691 EB 1115 EX DE,HL
E691 CD 96 E6 1116 LD A, (DE)
E691 EB 1117 EX DE,HL
E691 CD 96 E6 1118 LD A, (DE)
E691 EB 1119 EX DE,HL
E691 CD 96 E6 1120 LD A, (DE)
E691 EB 1121 EX DE,HL
E691 CD 96 E6 1122 LD A, (DE)
E691 EB 1123 EX DE,HL
E691 CD 96 E6 1124 LD A, (DE)
E691 EB 1125 EX DE,HL
E691 CD 96 E6 1126 LD A, (DE)
E691 EB 1127 EX DE,HL
E691 CD 96 E6 1128 LD A, (DE)
E691 EB 1129 EX DE,HL
E691 CD 96 E6 1130 LD A, (DE)
E691 EB 1131 EX DE,HL
E691 CD 96 E6 1132 LD A, (DE)
E691 EB 1133 EX DE,HL
E691 CD 96 E6 1134 LD A, (DE)
E691 EB 1135 EX DE,HL
E691 CD 96 E6 1136 LD A, (DE)
E691 EB 1137 EX DE,HL
E691 CD 96 E6 1138 LD A, (DE)
E691 EB 1139 EX DE,HL
E691 CD 96 E6 1140 LD A, (DE)
E691 EB 1141 EX DE,HL
E691 CD 96 E6 1142 LD A, (DE)
E691 EB 1143 EX DE,HL
E691 CD 96 E6 1144 LD A, (DE)
E691 EB 1145 EX DE,HL
E691 CD 96 E6 1146 LD A, (DE)
E691 EB 1147 EX DE,HL
E691 CD 96 E6 1148 LD A, (DE)
E691 EB 1149 EX DE,HL
E691 CD 96 E6 1150 LD A, (DE)
E691 EB 1151 EX DE,HL
E691 CD 96 E6 1152 LD A, (DE)
E691 EB 1153 EX DE,HL
E691 CD 96 E6 1154 LD A, (DE)
E691 EB 1155 EX DE,HL
E691 CD 96 E6 1156 LD A, (DE)
E691 EB 1157 EX DE,HL
E691 CD 96 E6 1158 LD A, (DE)
E691 EB 1159 EX DE,HL
E691 CD 96 E6 1160 LD A, (DE)
E691 EB 1161 EX DE,HL
E691 CD 96 E6 1162 LD A, (DE)
E691 EB 1163 EX DE,HL
E691 CD 96 E6 1164 LD A, (DE)
E691 EB 1165 EX DE,HL
E691 CD 96 E6 1166 LD A, (DE)
E691 EB 1167 EX DE,HL
E691 CD 96 E6 1168 LD A, (DE)
E691 EB 1169 EX DE,HL
E691 CD 96 E6 1170 LD A, (DE)
E691 EB 1171 EX DE,HL
E691 CD 96 E6 1172 LD A, (DE)
E691 EB 1173 EX DE,HL
E691 CD 96 E6 1174 LD A, (DE)
E691 EB 1175 EX DE,HL
E691 CD 96 E6 1176 LD A, (DE)
E691 EB 1177 EX DE,HL
E691 CD 96 E6 1178 LD A, (DE)
E691 EB 1179 EX DE,HL
E691 CD 96 E6 1180 LD A, (DE)
E691 EB 1181 EX DE,HL
E691 CD 96 E6 1182 LD A, (DE)
E691 EB 1183 EX DE,HL
E691 CD 96 E6 1184 LD A, (DE)
E691 EB 1185 EX DE,HL
E691 CD 96 E6 1186 LD A, (DE)
E691 EB 1187 EX DE,HL
E691 CD 96 E6 1188 LD A, (DE)
E691 EB 1189 EX DE,HL
E691 CD 96 E6 1190 LD A, (DE)
E691 EB 1191 EX DE,HL
E691 CD 96 E6 1192 LD A, (DE)
E691 EB 1193 EX DE,HL
E691 CD 96 E6 1194 LD A, (DE)
E691 EB 1195 EX DE,HL
E691 CD 96 E6 1196 LD A, (DE)
E691 EB 1197 EX DE,HL
E691 CD 96 E6 1198 LD A, (DE)
E691 EB 1199 EX DE,HL
E691 CD 96 E6 1200 LD A, (DE)
E691 EB 1201 EX DE,HL
E691 CD 96 E6 1202 LD A, (DE)
E691 EB 1203 EX DE,HL
E691 CD 96 E6 1204 LD A, (DE)
E691 EB 1205 EX DE,HL
E691 CD 96 E6 1206 LD A, (DE)
E691 EB 1207 EX DE,HL
E691 CD 96 E6 1208 LD A, (DE)
E691 EB 1209 EX DE,HL
E691 CD 96 E6 1210 LD A, (DE)
E691 EB 1211 EX DE,HL
E691 CD 96 E6 1212 LD A, (DE)
E691 EB 1213 EX DE,HL
E691 CD 96 E6 1214 LD A, (DE)
E691 EB 1215 EX DE,HL
E691 CD 96 E6 1216 LD A, (DE)
E691 EB 1217 EX DE,HL
E691 CD 96 E6 1218 LD A, (DE)
E691 EB 1219 EX DE,HL
E691 CD 96 E6 1220 LD A, (DE)
E691 EB 1221 EX DE,HL
E691 CD 96 E6 1222 LD A, (DE)
E691 EB 1223 EX DE,HL
E691 CD 96 E6 1224 LD A, (DE)
E691 EB 1225 EX DE,HL
E691 CD 96 E6 1226 LD A, (DE)
E691 EB 1227 EX DE,HL
E691 CD 96 E6 1228 LD A, (DE)
E691 EB 1229 EX DE,HL
E691 CD 96 E6 1230 LD A, (DE)
E691 EB 1231 EX DE,HL
E691 CD 96 E6 1232 LD A, (DE)
E691 EB 1233 EX DE,HL
E691 CD 96 E6 1234 LD A, (DE)
E691 EB 1235 EX DE,HL
E691 CD 96 E6 1236 LD A, (DE)
E691 EB 1237 EX DE,HL
E691 CD 96 E6 1238 LD A, (DE)
E691 EB 1239 EX DE,HL
E691 CD 96 E6 1240 LD A, (DE)
E691 EB 1241 EX DE,HL
E691 CD 96 E6 1242 LD A, (DE)
E691 EB 1243 EX DE,HL
E691 CD 96 E6 1244 LD A, (DE)
E691 EB 1245 EX DE,HL
E691 CD 96 E6 1246 LD A, (DE)
E691 EB 1247 EX DE,HL
E691 CD 96 E6 1248 LD A, (DE)
E691 EB 1249 EX DE,HL
E691 CD 96 E6 1250 LD A, (DE)
E691 EB 1251 EX DE,HL
E691 CD 96 E6 1252 LD A, (DE)
E691 EB 1253 EX DE,HL
E691 CD 96 E6 1254 LD A, (DE)
E691 EB 1255 EX DE,HL
E691 CD 96 E6 1256 LD A, (DE)
E691 EB 1257 EX DE,HL
E691 CD 96 E6 1258 LD A, (DE)
E691 EB 1259 EX DE,HL
E691 CD 96 E6 1260 LD A, (DE)
E691 EB 1261 EX DE,HL
E691 CD 96 E6 1262 LD A, (DE)
E691 EB 1263 EX DE,HL
E691 CD 96 E6 1264 LD A, (DE)
E691 EB 1265 EX DE,HL
E691 CD 96 E6 1266 LD A, (DE)
E691 EB 1267 EX DE,HL
E691 CD 96 E6 1268 LD A, (DE)
E691 EB 1269 EX DE,HL
E691 CD 96 E6 1270 LD A, (DE)
E691 EB 1271 EX DE,HL
E691 CD 96 E6 1272 LD A, (DE)
E691 EB 1273 EX DE,HL
E691 CD 96 E6 1274 LD A, (DE)
E691 EB 1275 EX DE,HL
E691 CD 96 E6 1276 LD A, (DE)
E691 EB 1277 EX DE,HL
E691 CD 96 E6 1278 LD A, (DE)
E691 EB 1279 EX DE,HL
E691 CD 96 E6 1280 LD A, (DE)
E69
```

E74D 21 DA EC	1042	LD HL,&TR4	E878	1188	E969 23	1336	INC HL
E750 CD EE E6	1043	CALL DNST	E878	1189	E96A 22 93 E0	1337	LD (#REC),HL
E753 01 F9 0F	1044	LD BC,\$FF9	E878	1190	E96D C9	1338	RET
E756 7E	1045	LD A,(HL);現在のTRACK	E878	1191	E96E	1339	
E757 ED 79	1046	OUT (C),A;	E878	1192	E96E	1340	■バンクからREAD
E759 3A A7 E0	1047	LD A,(\$SKCMD1);COMMAND	E878	1193	E96E CD B3 E6	1341	BFINS
E75C 57	1048	LD D,A	E878	1194	E971 D9	1342	CALL BG10MM
E75D 3A 80 EB	1049	LD A,(TRN)	E878	1195	E972 CD 7A E9	1343	EXX
E760 CD 9A E7	1050	CALL TRLP5;実行	E878	1196	E975 D9	1344	CALL DVINS2
E763 D8	1051	RET C	E878	1197	E976 C9	1345	EXX
E764 21 DA EC	1052	LD HL,&TR4	E878	1198	E977 CD AB E6	1346	RET
E767 CD EE E6	1053	CALL DNST	E878	1199	E977	1347	DVINS
E76A 3A 80 EB	1054	LD A,(TRN)	E878	1200	E977 CD AB E6	1348	CALL BGDN
E76D 77	1055	LD (HL),A;WORK更新	E878	1201	E97A 0A	1349	DVINS2
E76E C9	1056	RET	E878	1202	E97B D2 D9 E6	1350	LD A,(BC)
E76F	1057		E878	1203	E97D ED 78	1351	JP NC,BORST;IF NOT MEN
E76F	1058	SEEK;通常のSEEK	E878	1204	E97E ED 78	1352	IN A,(C)
E76F 21 DA EC	1059	LD HL,&TR4	E878	1205	E980 C3 D9 E6	1353	JP BGRST
E772 06 00	1060	LD B,0	E878	1206	E983	1354	
E774 3A 8B E0	1061	LD C,\$F5D	E878	1207	E983	1355	■バンクへWRITE
E777 4F	1062	LD C,A	E878	1208	E983	1356	BEFUS
E778 09	1063	ADD HL,BC	E878	1209	E983 08	1357	EX AP,AF'
E779 01 F9 0F	1064	LD BC,\$FF9	E878	1210	E983 CD B3 E6	1358	CALL BG10MM
E77C 7E	1065	LD B,0	E878	1211	E987 D9	1359	EXX
E77D ED 79	1066	OUT (C),A	E878	1212	E988 CD 91 E9	1360	CALL DVOU52
E77F 3A 80 EB	1067	LD A,(TRN)	E878	1213	E988 D9	1361	EXX
E782 BE	1068	CP (HL)	E878	1214	E98C C9	1362	RET
E783 C8	1069	RET 2;IF 同一TACK	E878	1215	E98D	1363	DVOUS
E784 3A A8 E0	1070	LD A,(\$SKCMD2);COMMAND	E878	1216	E98D 08	1364	EX AF,AF'
E787 57	1071	LD A,(HL);A;サイド	E878	1217	E98D CD AB E6	1365	CALL BGDN
E788 3A 80 EB	1072	LD A,(TRN)	E878	1218	E991	1366	DVOUS2
E78B CD 9A E7	1073	CALL TRLP5;実行	E878	1219	E991 38 05	1367	JP C,DVOUS3;IF MEM
E78E D8	1074	RET C	E878	1220	E993 08	1368	EX AF,AF'
E78F 21 DA EC	1075	LD HL,&TR4	E878	1221	E994 ED	1369	LD (BC),A
E792 CD EE E6	1076	CALL DNST	E878	1222	E995 C3 D9 E6	1370	JP BGRST
E795 3A 80 EB	1077	LD A,(TRN)	E878	1223	E998	1371	DVOUS3
E798 77	1078	LD (HL),A;WORK更新	E878	1224	E998 08	1372	EX AF,AF'
E799 C9	1079	RET	E878	1225	E999 ED 79	1373	OUT (C),A
E79A	1080		E878	1226	E99B C3 D9 E6	1374	JP BGRST
E79A	1081	TRLP5;A:TRACK D=COMMAND	E878	1227	E99E	1376	EMM 入出力
E79A 01 FB 0F	1082	LD BC,\$FFB	E878	1228	E99E	1377	
E79D ED 78	1083	OUT (C),A	E878	1229	E99E	1378	EMMRW
E79F 0E F8	1084	LD C,\$F5D	E878	1230	E99E 87	1379	ADD A,A;
E7A1 ED 51	1085	OUT (C),D	E878	1231	E99F 87	1380	ADD A,A;
E7A3 C3 0D EA	1086	JP WTP1 WAIT	E878	1232	E9A0 4F	1381	LD C,A;EMM?
E7A6	1087		E878	1233	E9A1 06 0D	1382	LD B,\$D
E7A6	1088	;(1レコード入出力)	E878	1234	E9A3 2A 93 E0	1383	LD HL,(\$REC)
E7A6	1089	DMAを使用	E878	1235	E9A6 ED 5B 9F E0	1384	LD DE,(\$SLNG)
E7A6	1090	RWREC;IN BF,SLNG,REC,LSCT,DN	E878	1236	E9A6 CD 44 E6	1385	CALL MLTHD
E7A6	1091	LD A,(\$DN)	E878	1237	E9AD ED 63 E6	1386	OUT (C),L;下位
E7A6	1092	CP 4	E878	1238	E9AF 03	1387	INC BC
E7A6	1093	JP NC,RAMRC;RAM-DISK	E878	1239	E9B6 ED 61	1388	OUT (C),H;中位
E7A6	1094		E878	1240	E9B2 83	1389	INC BC
E7A6	1095	LD A,(\$FRWF);FLOPPY-DISK	E878	1241	E9B3 C5	1390	PUSH BC
E7A6	1096	OR A	E878	1242	E9B4 2A 93 E0	1391	LD HL,(\$REC)
E7A6	1097	JP NZ,WTRC;WRITE	E878	1243	E9B7 11 00 01	1392	LD DE,\$5
E7A6	1098	LD HL,(\$SLNG)	E878	1244	E9B8 CD 44 E6	1393	CALL DIVHD
E7A6	1099	DEC HL	E878	1245	E9BD C1	1394	POP BC
E7A6	1100	LD (SLNG1),HL;セクター長	E878	1246	E9BE ED 79	1395	OUT (C),A;上位 ADR
E7A6	1101	LD HL,(\$BF)	E878	1247	E9C0 83	1396	INC BC
E7A6	1102	LD (BUFF1),HL;BUFF ADR	E878	1248	E9C1 C5	1397	PUSH BC
E7A6	1103	LD HL,RODMAD;DMA DATA	E878	1249	E9C2 D9	1398	EXX
E7A6	1104	LD A,\$80;READ COMMAND	E878	1250	E9C3 C1	1399	POP BC
E7A6	1105	JP SKPRC	E878	1251	E9C4 D9	1400	EXX
E7A6	1106		E878	1252	E9C5 21 67 E0	1401	LD HL,REMMDT
E7A6	1107	WTRC;WRITE REC	E878	1253	E9C8 3A A1 E0	1402	LD A,(\$FRWF)
E7A6	1108	LD HL,(\$SLNG)	E878	1254	E9CB 87	1403	OR A
E7A6	1109	LD HL,(\$SLNG2),HL	E878	1255	E9CC 28 04	1404	JR Z,EMMPH1;IF READ
E7A6	1110	LD HL,(\$BF)	E878	1256	E9CE 11 0C 00	1405	LD DE,12
E7A6	1111	LD HL,(\$BF)	E878	1257	E9D1 19	1406	ADD HL,DE;REMMDT+12
E7A6	1112	LD (BUFF2),HL	E878	1258	E9D2 CD B3 E6	1407	EMMPH1
E7A6	1113	LD HL,RODMAD;DMA DATA	E878	1259	E9D5 38 04	1408	CALL BG10MM
E7A6	1114	LD A,\$A0;WRITE COMMAND	E878	1260	E9D7 11 06 00	1409	JR Z,EMMPH2;IF GRAM
E7A6	1115	SKPRC;読み書き共通	E878	1261	E9DA 1D	1410	LD DE,12
E7A6	1116	LD (DMACMD),HL	E878	1262	E9DB	1411	ADD HL,DE;REMMDT+6,18
E7A6	1117	LD (FDCMD),A	E878	1263	E9DB	1412	EMMPH2
E7A6	1118	PUSH BC	E878	1264	E9DB 11 EB E9	1413	LD DE,EMMPH2
E7A6	1119	PUSH DE	E878	1265	E9DE 01 08 00	1414	LD BC,\$D
E7A6	1120	CALL GMCHG;バンク切り替えなど	E878	1266	E9E1 ED 78	1415	LD LR;自己書き換え
E7A6	1121	CALL RECTR;DATA SET	E878	1267	E9E3 ED 4B 91 E0	1416	LD RC,(\$BF);BUFF ADR
E7A6	1122	CALL NC,MTRON	E878	1268	E9E7 ED 5B 9F E0	1417	LD DE,(\$SLNG);セクター長
E7A6	1123	CALL NC,SEEK	E878	1269	E9EB	1418	EMMCLP
E7A6	1124	JP C,SSTPM	E878	1270	E9EB	1419	EMMPH
E7A6	1125	LD BC,\$FFA	E878	1271	E9F1 03	1420	DS 6;書き換え場所
E7A6	1126	LD A,(SCT)	E878	1272	E9F2 1B	1421	INC BC
E7A6	1127	OUT (C),A;セクター	E878	1273	E9F3 7A	1422	DEC DE
E7A6	1128	LD C,\$F8	E878	1274	E9F4 B3	1423	LD A,D
E7A6	1129	LD A,(FDCMD)	E878	1275	E9F5 28 F4	1424	OR E
E7A6	1130	OUT (C),A;FDC COMMAND	E878	1276	E9F7 ED 43 91 E0	1425	JP NZ,EMMCLP
E7A6	1131	LD HL,(\$DMACMD),HL	E878	1277	E9F8 2A 93 E0	1426	LD (BF),BC
E7A6	1132	CALL SETDMA	E878	1278	E9FE 23	1427	LD HL,(\$REC)
E7A6	1133	CALL WTP22;WAIT	E878	1279	E9FE 23	1428	INC HL
E7A6	1134	JP C,SSTPM	E878	1280	E9FE 22 93 E0	1429	LD (#REC),HL
E7A6	1135	LD A,\$83	E878	1281	E9A0 CD B3 E6	1430	CALL BGRST
E7A6	1136	LD BC,\$F80	E878	1282	E9A5 C3 31 E9	1431	JP EMMRT
E7A6	1137	OUT (C),A;RESTDMA	E878	1283	E9A8	1432	
E7A6	1138	LD HL,(\$BF);BUFF ADR	E878	1284	E9A8	1433	;(WAIT)
E7A6	1139	LD DE,(\$SLNG);セクター長	E878	1285	E9A8	1434	
E7A6	1140	ADD HL,DE;BF+BF+SLNG	E878	1286	E9A8	1435	WTRC;MOTOR用
E7A6	1141	LD (BF),HL	E878	1287	E9A8 16 81	1436	LD D,\$81
E7A6	1142	LD HL,(\$REC);レコード番号	E878	1288	E9A8 C3 14 EA	1437	JP WNBVS
E7A6	1143	INC HL;REC=REC+1	E878	1289	E9A8	1438	WTP1;READその他
E7A6	1144	LD (#REC),HL	E878	1290	E9A8 16 99	1439	LD D,\$99
E7A6	1145	LD BC,\$FFC	E878	1291	E9AF C3 14 EA	1440	JP WNBVS
E7A6	1146	LD A,(\$DN)	E878	1292	E9B2	1441	WTP22;WRITE用
E7A6	1147	OUT (C),A;MTRF	E878	1293	E9B2 16 FD	1442	LD D,\$FD
E7A6	1148	OR A	E878	1294	E9B3	1443	WNBVS
E7A6	1149	SSTPM	E878	1295	E9B3	1444	LD BC,\$FF8
E7A6	1150	LD BC,\$F80	E878	1296	E9B3 01 F8 0F	1445	LD A,(\$WAIT)
E7A6	1151	LD A,(\$F80D)	E878	1297	E9B3 3A AC E0	1446	LD E,A
E7A6	1152	OUT (C),A	E878	1298	E9B3 57	1447	WAITL2
E7A6	1153	LD BC,\$800	E878	1299	E9B3 CD B3 E6	1448	LD HL,\$1000;WAIT 関
E7A6	1154	LD A,\$10	E878	1300	E9B3	1449	WNBVS
E7A6	1155	OUT (C),A	E878	1301	E9B3 2B	1450	DEC HL
E7A6	1156	EI	E878	1302	E9B3 7C	1451	LD A,H
E7A6	1157	POP DE	E878	1303	E9B3 7C	1452	OR L
E7A6	1158	POP BC	E878	1304	E9B3 7C	1453	JP NZ,WNBLS2
E7A6	1159	RET	E878	1305	E9B3 7C	1454	DEC E
E7A6	1160		E878	1306	E9B3 7C	1455	JP NZ,WNBLS2
E7A6	1161	WRITE TRACK(FORMAT専用)	E878	1307	E9B3 7C	1456	JP DSTOP;エラー
E7A6	1162	WTRC;構造は同じ	E878	1308	E9B3 7C	1457	WNBLS2
E7A6	1163	PUSH BC	E878	1309	E9B3 7C	1458	LD A,(C)
E7A6	1164	PUSH DE	E878	1310	E9B3 7C	1459	AND D
E7A6	1165	LD HL,(\$SLNG)	E878	1311	E9B3 7C	1460	JP NZ,WNBVS
E7A6	1166	DEC HL	E878	1312	E9B3 7C	1461	OR A
E7A6	1167	LD (SLNG2),HL	E878	1313	E9B3 7C	1462	RET
E7A6	1168	LD HL,(\$BF)	E878	1314	E9B3 7C	1463	DSTOP;エラー
E7A6	1169	LD (BUFF2),HL	E878	1315	E9B3 7C	1464	IN A,(C)
E7A6	1170	LD HL,&GMCHG	E878	1316	E9B3 7C	1465	AND D
E7A6	1171	CALL RECTR	E878	1317	E9B3 7C	1466	LD C,A
E7A6	1172	CALL NC,MTRON	E878	1318	E9B3 7C	1467	AND \$10
E7A6	1173	CALL NC,SEEK	E878	1319	E9B3 7C	1468	CALL NZ,ERR7
E7A6	1174	LD HL,RODMAD	E878	1320	E9B3 7C	1469	LD A,C
E7A6	1175	LD HL,(\$F80D)	E878	1321	E9B3 7C	1470	LD A,C
E7A6	1176	CALL SETDMA	E878	1322	E9B3 7C	1471	LD A,C
E7A6	1177	LD BC,\$FF8	E878	1323	E9B3 7C	1472	CALL Z,ERR2
E7A6	1178	LD A,\$F8	E878	1324	E9B3 7C	1473	ERR2
E7A6	1179	OUT (C),A;FDC COMMAND	E878	1325	E9B3 7C	1474	LD A,(\$DN)
E7A6	1180	CALL WTP22	E878	1326	E9B3 7C	1475	LD BC,\$FFC
E7A6	1181	CALL NC,RESTDMA	E878	1327	E9B3 7C	1476	OUT (C),A;MOTOR OFF
E7A6	1182	CALL NC,MTRF	E878	1328	E9B3 7C	1477	CALL RESTDMA;RESET DMA
E7A6	1183	SSTPM	E878	1329	E9B3 7C	1478	LD HL,\$0
E7A6	1184	CALL BGRST	E878	1330	E9B3 7C	1479	LD (#RCB),HL
E7A6	1185	POP DE	E878	1331	E9B3 7C	1480	LD (\$C3B),HL
E7A6	1186	POP BC	E878	1332	E9B3 7C	1481	SCF;エラーはCARRY FLAG!
E7A6	1187	RET	E878	1333	E9B3 7C	1482	RET
E7A6	1188		E878	1334	E9B3 7C		
E7A6	1189		E878	1335	E9B3 7C		
E7A6	1190		E878	1336	E9B3 7C		
E7A6	1191		E878	1337	E9B3 7C		
E7A6	1192		E878	1338	E9B3 7C		
E7A6	1193		E878	1339	E9B3 7C		
E7A6	1194		E878	1340	E9B3 7C		
E7A6	1195		E878	1341	E9B3 7C		

EA58	1483	:	IOHMEM1:\$10->\$10	FE08	1758	SSTPMH
EA58	1484	ERR0	DB \$14,\$28,\$80,\$8D,\$FB,\$0F,\$0	FE08 01 D0 IF	1759	LD BC,\$1FD0 ;
EA58 AF	1485	XOR A	2,\$CF,\$05,\$CF,\$87	FE08 3A A2 E0	1760	LD A,(\$F0D0) ;
EA59 32 8C E0	1486	LD A,(\$STOP),A		FE08 ED 79	1761	OUT (C),A ; 画面回り
EA5C C9	1487	RET		FE08 01 00 0B	1762	LD BC,\$800 ;
EA5D	1488	ERR1		FE03 3E 18	1763	LD A,\$10 ;
EA5D 3E 01	1489	LD A,1		FE05 ED 79	1764	OUT (C),A ; BANK
EA5F 18 1A	1490	JR ERR		FE07 FB	1765	EX
EA61	1491	ERR2		FE08 D1	1766	POP DE
EA61 3E 02	1492	LD A,2		FE09 C1	1767	POP BC
EA63 18 16	1493	JR ERR		FE0A C9	1768	RET
EA65	1494	ERR3		FE0B	1769	
EA65 3E 03	1495	LD A,3		FE0B	1770	; ディスクから
EA67 18 12	1496	JR ERR		FE0B	1771	; 読み込み
EA69	1497	ERR4		FE0B	1772	REDI ;
EA69 3E 04	1498	LD A,4		FE0B ED 78	1773	IN A,(C) ; STATUS
EA6D 18 0E	1499	JR ERR		FE0D 0F	1774	RRCA
EA6D 3E 05	1500	ERR5		FE0E D0	1775	RET NC
EA6F 18 0A	1501	LD A,5		FE0F 0F	1776	RRCA
EA71	1502	JR ERR		FE0A 38 F9	1777	JR NC,REDI
EA71 3E 06	1503	ERR6		FE0A 24 A	1778	LD C,D ; DATAレジスタ
EA73 18 0E	1504	LD A,6		FE0A ED 78	1779	IN A,(C) ; 下位8BIT
EA75	1505	JR ERR		FE0A D9	1780	EXX
EA75 3E 07	1506	ERR7		FE0A	1781	REDPAC
EA77 18 02	1507	LD A,7		FE0A ED 79	1782	OUT (C),A ; LD (C),A
EA79 3E 09	1508	JR ERR		FE0A 83	1783	INC BC
EA7B	1509	ERR9		FE0A D9	1784	EXX
EA7B 32 8C E0	1510	LD A,9		FE0A 4B	1785	LD C,E ; STATUSレジスタ
EA7E 37	1511	RET		FE0A 18 E	1786	JR RED1
EA7F C9	1512	LD (\$STOP),A		FE0A	1787	
EA80	1513	SCF		FE0A	1788	
EA80	1514	RET		FE0A	1789	WTREG
EA80	1515			FE0A F3	1790	DI
EA80	1516	; RESET DMA		FE0A CD B3 E6	1791	CALL BG10MM
EA80	1517	RESTDMA		FE0B 21 ED 78	1792	LD HL,\$78ED ; IN A,(C)
EA80 3E 83	1518	LD A,\$83		FE0A 38 03	1793	JR C,PACX2
EA82 01 80 1F	1519	LD BC,\$1F80		FE05 21 0A 00	1794	LD HL,\$800A ; LD A,(BC)
EA85 ED 79	1520	OUT (C),A		FE0B	1795	PACX2
EA87 C9	1521	RET		FE0B 22 DF F0	1796	LD (WRTPAC),HL
EA88	1522			FE0C 22 C0 F0	1797	LD (WRTPAC2),HL
EA88	1523	; DMA COMMANDをSET		FE0F D9	1798	
EA88	1524	SETDMA		FE0C9	1799	WRTPAC2
EA88 01 80 1F	1525	LD BC,\$1F80		FE0C ED 78	1800	IN A,(C)
EA8B 5E	1526	LD E,(HL) ; (HL+1)		FE0C 08	1801	EX AF,AF'
EA8C 23	1527	INC HL ; からの(HL)画		FE0C D9	1802	EXX
EA8D	1528	STDMALP		FE0A 3E A0	1803	LD A,\$A0 ; コマンド WRITE
EA8D 04	1529	INC B		FE0C CD E6 F0	1804	CALL SETSCP ; READY
EA8E ED A3	1530	OUT I		FE0C DA 88 F0	1805	JP C,SSTPMH
EA90 1D	1531	DEC E		FE0C CD D2 F0	1806	CALL WRTI ; 書き込み
EA91 20 FA	1532	JR NZ,STDMALP		FE0C 34 47 F0	1807	JP RNTOG
EA93 C9	1533	RET		FE0D	1808	
EA94	1534			FE0D	1809	WRTI
EA94	1535	; 文字列中、改行		FE0D ED 78	1810	IN A,(C) ;
EA94	1536	; まで切り出す		FE0A 8F	1811	RRCA ; STATUS
EA94	1537	; 独立ルーチン		FE0D D8	1812	RET NC ;
EA94	1538			FE0D 0F	1813	RRCA ;
EA94	1539	GYOKI		FE0D 30 F9	1814	JR NC,WRTI
EA94 01 00 FE	1540	LD BC,\$F800		FE0D 4A	1815	LD C,D ; DATA
EA97 ED 5D 9D E0	1541	LD DE,(\$NSET)		FE0A 08	1816	EX AF,AF'
EA9B 2A 91 E0	1542	LD HL,(\$BF)		FE0D ED 79	1817	OUT (C),A
EA9E DD 21 02 EE	1543	LD IX,P256+2		FE0D D9	1818	EXX
EA9A 7A	1544	GYOKILP		FE0D 03	1819	INC BC
EA9A 83	1545	LD A,D ; 文字列終わり?		FE0F	1820	WRTPAC
EA9A 20 C	1546	OR E		FE0F ED 78	1821	IN A,(C)
EA9A 3A E7 EC	1547	JR NZ,GYOKISK2 ; IF NOT END		FE0E 08	1822	EX AF,AF'
EA9A DD BE FF	1548	LD A,(\$OF2)		FE0E D9	1823	EXX
EA9C 20 01	1549	CP (IX+9FF)		FE0E 4B	1824	LD C,E ; STATUS
EA9E 0D	1550	JR NZ,GYOFOFSK		FE0E 18 EC	1825	JR WRTI
EA9F	1551	DEC C ; EOFを抜く		FE0E	1826	
EA9F	1552	GYOFOFSK		FE0E	1827	READY
EA9F AF	1553	XOR A		FE0E	1828	
EA9D 18 17	1554	JR GYOKISK2		FE0E 32 8C EB	1829	SETSCP ; IN A
EA9D	1555	GYOKISK		FE0E C5	1830	LD (FDCMD),A
EA9D 1B	1556	DEC DE		FE0E D5	1831	PUSH DE
EA9D CD 7A E6	1557	CALL LDAHL		FE0E CD FC E6	1832	PUSH DE
EA9D 23	1558	INC HL		FE0E DA 6F E7	1833	CALL MTRON
EA9D FE 0D	1559	OR E		FE0F D1	1834	CALL NC,SEEK
EA9D 20 C	1560	JR NZ,GYONEND		FE0F C1	1835	POP DE
EA9D 3E 02	1561	LD A,2		FE0F D8	1836	POP BC
EA9D 18 0A	1562	JR GYOKISK2		FE0F 3D	1837	RET
EA9F	1563	GYONEND		FE0F 81 F8 0F	1838	LD BC,\$FF8
EA9F DD 77 00	1564	LD (IX+0),A ; (IX)-(-HL)		FE0F 3A 8C EB	1839	LD A,(FDCMD)
EA9D 23	1565	INC IX		FE0F 3E 07	1840	OUT (C),A ; コマンドレジスタ
EA9D 0C	1566	INC C		FE0F	1841	LD A,7 ;
EA9D 18 DB	1567	DJNZ GYOKILP		FE0F 3D	1842	SETSCP ;
EA9D 3E 01	1568	LD A,1		FE0F 3D	1843	DEC A ; WAIT
EA9D	1569	GYOKISK2		FE0F 20 FD	1844	JR NZ,SETSCP
EA9D 32 00 EE	1570	LD (P256),A ; 改行 FLUG		FE0F C9	1845	RET
EA9D 3E 01	1571	LD A,C ; 文字数		FE0F	1846	
EA9D 32 01 EE	1572	LD (P256+1),A		FE0F	1847	;/ WRITE TRACK /
EA9D 22 91 E0	1573	LD (\$BF),HL		FE0F	1848	(non DMA)
EA9D ED 53 9D E0	1574	LD (\$MBT),DE		FE0F	1849	X22
EA9D C9	1575	RET		FE0F C5	1850	PUSH BC
EA9D	1576			FE0F D5	1851	PUSH DE
EA9D	1577	; フルバス作成		FE0F CD 78 E8	1852	CALL RECTRN
EA9D	1578			FE0F DA FC E6	1853	CALL NC,MTRON
EA9D 179 SPATH	1579	LD A,(\$SPATH),B		FE0F DA 4D E7	1854	CALL NC,SEEK
EA9D 2A A9 E0	1580	LD HL,(\$SPATH)		FE0F 38 48	1855	JR C,SSTPMH ; どの値も
EA9D DD 21 81 EE	1581	LD IX,P256+981		FE0F 3E 03	1856	LD A,3 ;
EA9D AF	1582	XOR A ; 文字列長		FE0F 32 5D F1	1857	LD (RTRY),A ; 3度まで
EA9D 32 80 EE	1583	LD A,(\$80),A		FE0F F3	1858	DI
EA9D 3E 04	1584	LD C,A		FE0F CD E6	1859	CALL BG10MM
EA9D	1585	SPATHLP		FE0F B3 ED 78	1860	LD HL,\$78ED ; IN A,(C)
EA9D 7E	1586	LD A,(HL)		FE0F 38 03	1861	JR C,PACX3
EA9D 23	1587	INC HL		FE0F 21 0A 00	1862	LD HL,\$800A ; LD A,(BC)
EA9D B7	1588	OR A		FE0F	1863	PACX3
EA9D CA 65 EA	1589	JP Z,ERR3 ; END OF PATH		FE0F 22 DF F0	1864	LD (WRTPAC),HL
EA9D FE 3B	1590	CP ' ; PATHの区切り		FE0F 22 2F F1	1865	LD (WRTPAC3),HL
EA9D 28 0E	1591	JR Z,SPATHS		FE0F	1866	RTRYLP
EA9D DD 77 00	1592	LD (IX+0),A ; 下位DIR		FE0F 16 FB	1867	LD D,\$FB ; STATUS
EA9D 23	1593	INC IX		FE0F 18 F8	1868	LD E,\$FB ; DATA
EA9D 8C	1594	INC C		FE0F D9	1869	EXX
EA9D 18 EE	1595	JR SPATHLP		FE0F ED 4B 91 E0	1870	LD BC,(\$BF) ; BUFF ADR
EA9D	1596	SPATHSK		FE0F	1871	WRTPAC3
EA9D 1A	1597	LD A,(DE)		FE0F ED 78	1872	IN A,(C)
EA9D 13	1598	INC DE		FE0F	1873	EX AF,AF'
EA9D DD 77 00	1599	LD (IX+0),A ; NAME		FE0F D9	1874	
EA9D DD 23	1600	INC IX		FE0F 01 F8 0F	1875	LD BC,\$FF8 ; FDC COMMAND
EA9D C	1601	INC C		FE0F 3E F0	1876	LD A,\$F0 ; WRITE TRACK
EA9D 18 F6	1602	LD A,C ; 文字列長		FE0F ED 79	1877	OUT (C),A ;
EA9D 79	1603	LD (P256+\$80),A		FE0F 3E 07	1878	LD A,3 ; WAIT
EA9D 22 A9 E0	1604	LD (\$BPATH),HL		FE0F	1879	SETSCP2 ;
EA9D B7	1605	OR A		FE0F 3D	1880	DEC A ;
EA9D C9	1606	RET		FE0F 3D FD	1881	JR NZ,SETSCP2
EA9D	1607			FE0F CD D2 F0	1882	CALL WRTI ; 書き込み
EA9D	1608			FE0F 42 07	1883	RRCA
EA9D	1609	; DMA DATA /		FE0F 4E D6	1884	AND \$DF
EA9D	1610			FE0F 45 28 EC	1885	JR Z,NDSTOP2
EA9D	1611	RDMDM ; READ		FE0F 21 5D F1	1886	LD HL,RTRY ; RETRY
EA9D 0F 83 7D FB	1612	LD \$15,\$83,\$7D,\$FB,\$0F		FE0F 44 35	1887	DEC HL ;
EA9D 0F	1613	SLNG1		FE0F C2 26 F1	1888	JP NZ,RTRYLP ; ERR
EA9D FE 00	1614	DW 256-1		FE0F CD 31 EA	1889	CALL DSTOP ; ERR
EA9D 2C	1615	LD \$2C		FE0F 18 04	1890	JR SSTPMH
EA9D	1616	IOHMEM1:\$10->\$18		FE0F	1891	NDSTOP2
EA9D 10 80 8D	1617	LD \$10,\$80,\$8D		FE0F CD 1F E7	1892	CALL NTROP
EA9D	1618	BUFF1		FE0F B7	1893	OR A
EA9D 00 00	1619	DW 0		FE0F	1894	SSTPMH
EA9D 92 CF 87	1620	LD \$92,\$CF,\$87		FE0F CD D9 E6	1895	LD HL,NDST ; 元へ戻す
EA9D	1621			FE0F D1	1896	POP DE
EA9D	1622	WTDMD ; WRITE		FE0F C1	1897	POP BC
EA9D 11 83 79	1623	DW 17,\$83,\$79		FE0F C9	1898	RET
EA9D	1624	RUFF2		FE0F	1899	
EA9D 00 00	1625	DW 0		FE0F	1900	RTRY DB 0
EA9D	1626	SLNG2		FE0F	1901	
EA9D FF 00	1627	DW 256-1		FE0F	1902	

▶「20世紀楽ドリル」もすごかった。「気分転換V」もすごかった。しかし、それをはるかに超える代物を知ってしまいました。それはUCCの「ワークス」シリーズです。なかでも「ハードワーク」は柿エキスイ入りのものでそれはもう……フタを開けた瞬間に死ぬます。ほかの2種もはっきり言って毒物です。

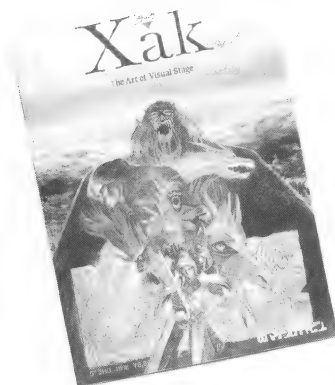
升井 晋也 (21) 福岡県

愛読者プレゼント

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ、希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1990年7月18日の到着分までとします。当選者の発表は1990年9月号で行います。

マイクロキャビン ☎0593(51)6482



2

サーク

X68000用 5"2HD版 4 枚組
8,800円 3名

ひさびさにX68000に新作を出したマイクロキャビンの自信作。主人公の青年が、悪の親玉バドゥーを封印するため旅に出るというRPG。

シャープ ☎03(260)1161

1

ダウントウン 熱血物語

X68000用 5"2HD版 2 枚組
8,800円 3名



ファミコンでもお馴染みの熱血シリーズ移植第2弾。バリバリの硬派、くにお君とりき君が町じゅうを駆けめぐり大活躍する、ケンカアクションゲームだ。

日本ファルコム ☎0425(27)0555

4

ファルコム システム手帳

2名

ワンダラーズ・フロム・イースが評判の日本ファルコムから、システム手帳をプレゼント。2タイプ各1名ずつ、どちらが当たるかは着いてのお楽しみ!

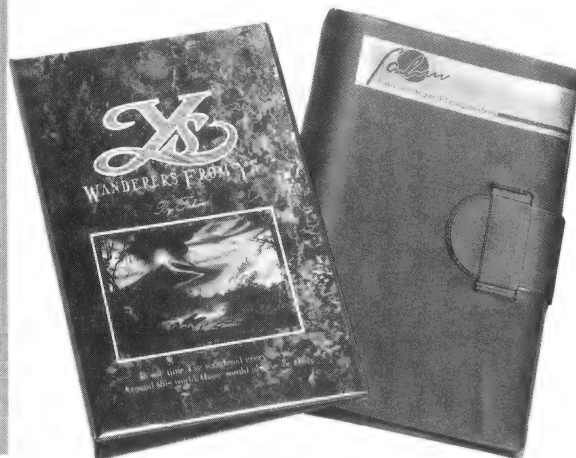
システムソフト ☎092(752)3902

3

天下統一

X68000用 5"2HD版 2 枚組
9,800円 3名

大戦略などでお馴染みのシステムソフトの最新戦国シミュレーションゲーム。移植は同じ九州のよしみか、あのアルシスソフトが担当している。



日本ソフトバンクの メモ帳

5名

5

Oh!X編集部で用達のメモ帳。このメモ帳、システム手帳にもはさめるというスグレモノ。5名の方にプレゼントしちゃいます。



5月号プレゼント当選者

- ①ポピュラス (秋田県) 田口勝己 (滋賀県) 森夕香 (奈良県) 澤克彦
②バブルポブル (東京都) 白水健太 (茨城県) 唯野晃 (兵庫県) 木村健二
③リウィード (北海道) 渡部一郎 (千葉県) 中村敏祐 (愛知県) 白井達広
④ファルコムグッズ A (静岡県) 渡辺直樹 (鹿児島県) 本真光
B (秋田県) 尾形淳一 (東京都) 阿部亨 (長野県) 塚本隆司 C (奈良県) 川上峰且 (兵庫県) 石崎義忠 D (青森県) 沢田優 (宮城県) 小野寺光 (大阪府) 堤博之 E (千葉県) 大古哲生 (香川県) 森洋幸
⑤高麗人参飴 (茨城県) 安部博幸

以上の方々が当選されました。おめでとうございます。商品は順次発送いたしますが、入荷状況などにより遅れる場合もあります。また、公正取引委員会の告示により、このプレゼントに当選された方は、この号の他の懸賞には当選できない場合がありますのでご了承ください。

(価格はすべて消費税別です)

郵便はがき

料金受取人払

1 0 2

麴町局
承認

3018

差出有効期間

1990年 9 月 30

日まで

(受取人)

東京都千代田区

九段南2-3-26井関ビル

(株)日本ソフトバンク

Oh!  編集室行

□□□-□□

電話

住所

氏名

年齢

職業・勤務先
学校・学部・学年

●編集室へのメッセージ

今月号の特集について	
いちばん良かった記事	興味のなかった記事
これから載せてほしい記事内容	本誌以外にお読みのパソコン雑誌
推薦する市販ソフト	
ソフト名:	
推薦理由:	
現在のワープロに対する不満点を挙げてください	
あなたの愛機は(所有機種に○印をつけてください) ない X1(マニアタイプ,C,D,F,G,twin) X1turbo(model 10,20,30,40,II,III,Z,ZII,ZIII) MZ-(80K/C, 1200, 700, 1500, 80B, 2000, 2200, 2500, 2861) X68000(無印,ACE,PRO,PRO II,EXPERT,EXPERT II,SUPER,HD) その他 FD(基) TAPE QD HD(MB) プリンタ()	
年齢	歳
パソコン歴	年
男・女	プレゼントNo.

振替用紙

点線からきれいに切り取ってご使用いただけます。

通常払込料金
加入者負担

払込票

郵便番号	東京	1	2	9	3	0	7
加入者名	株式会社 日本ソフトバンク						
全額	使	千	百	十	千	百	十
払込人住所氏名							
備考							
受付局日付印							

切り取らないで郵便局にお出してください。

記載事項を訂正した場合は、その箇所に訂正印を押してください。

通常払込料金
加入者負担

払込通知票

郵便番号	東京	1	2	9	3	0	7
加入者名	株式会社 日本ソフトバンク						
払込人住所氏名							
料	金	備	考	受付局日付印			
払込み	特	外	司				

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。（郵 政 省）

各票の※印欄は、払込人において記載してください。

送り先

お名前	フリカナ	性別	男・女	年齢		ご職業	
ご住所	フリカナ						
〒							
お電話	二	口	宅	勤	務	先	

申込書

THE COMPUTER 定期購読	新規申し込み 年 月 号より	継続申し込み TC NO.	年 間 7,200円
Oh./PC 定期購読	新規申し込み 年 月 号より	継続申し込み PC NO.	年 間 11,440円 6ヶ月 5,720円
Oh./X 定期購読	新規申し込み 年 月 号より	継続申し込み X NO.	年 間 6,720円
Oh./FM 定期購読	新規申し込み 年 月 号より	継続申し込み FM NO.	年 間 6,720円
月刊情報 定期購読	新規申し込み 年 月 号より	継続申し込み JS NO.	年 間 8,160円 6ヶ月 4,080円
C MAGAZINE 定期購読	新規申し込み 年 月 号より	継続申し込み CM NO.	年 間 11,760円
パソコン 定期購読	新規申し込み 年 月 号より	継続申し込み PM NO.	年 間 6,960円
Beep メガドライブ 定期購読	新規申し込み 年 月 号より		年 間 5,760円

通信欄

この欄は、加入者あての通信にお使いください。
切り取らないで郵便局にお出しください。



この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。(郵 政 省)

[第2話]

本命は恐ろしい

TAKAHARA HIDEKI 高原 秀己

競馬を楽しむ女性が増えている。

純粋にギャンブルとして興じるいわゆる「オジギャル」をはじめ、タケ・ユタカ人氣に乗り遅れまいとする女の子、果てはデートコースとして馬券を買う彼にくっついてくる彼女、とパターンはさまざまだが、本当に競馬場でも場外馬券売り場でも女性の姿が目立って多くなった。

券を買うわけでもないのに彼にくっついてきて長蛇の列をさらに長くする女の子にはご遠慮いただきたいが、殺伐としたギャンブルの場に女性の姿が目立つ光景はいいものだ。

もともと華やかな競馬。見る側にファッショナブルな要素が加わって、オシャレなギャンブルとして人氣急上昇。一種のブーム現象にまでなっているといっている。

沸騰したのが5月27日に開催された「競馬の祭典」日本ダービーだ。東京競馬場の入場者19万人、JRAの売り上げ397億円といずれも競馬史上最高の活況にわいた。

4歳サラブレッド馬だけの戦いとあって、ほかのビッグレースとはひと味違う趣がブンプン。レースも最後まで大混戦となり、3番人氣のアイネスフウジンが新記録タイムで今年のダービー馬に輝いた。武豊騎乗の人氣馬ハクタイセイはゴール間近で足が伸びず、2着には本命のメジロライアンが入った。

連勝複式は3-5で、7.7倍。けっこう好配当だが、これ、2番目に人氣があった組み合わせ。あとはすべて10倍以上のオッズだったから、大本命みたいなもの。穴を狙ったばかりも押さえに少しだけ買っておいたので、差し引きで若干のプラスになった。

それにしても、ビッグレース続きだったこの春の競馬。天皇賞からダービーまで毎週毎週、よくまあこれだけ飽きもせず本命格の馬ばかりが入ったもんだ。天皇賞のスーパークリーク、安田記念のオグリキャップはいうに及ばず、桜花賞、さつき賞、ダービーと終わってみればことごとく人氣上位3頭のどれかが軒並み1着、2着を占めている。

よく飲みに行く店の常連サンに、

「競馬は夢を買うもんや」

という哲学を持った危ない風貌のおニイサンがよく来る。何百倍という大穴ばかりを狙い続けているが、戦績はサッパリ。こうなると、救いがない。キツチリと本命に賭ける地味な生き方のオジサンは逆に毎週連勝に次ぐ連勝。

ところで競馬の本命優位は一時的な現象なのかもしれないが、世の中全般にわたり、最近、まず「大穴」はこない。「予想外のハプニング！」なんてことはついぞお目にかからない。

前回は「選択の時代」という歓迎できる風潮になってきたことを書いたが、これは一面では自民党対社会党、トヨタ対日産といった感じの複数の大手が拮抗してそのフィールドを支配してしまうという大いに嘆くべき時代の到来を意味しているのかもしれないのだ。

プロ野球で考えてみると、はっきりわかるが、西武と巨人が独走していることに対し、奇妙な安心感が漂っている（ばく自身は巨人がダントツの実力を持っているとは思っていないが、カーブと並んで戦力が整っていることだけは認める）。阪神やダイエーが無人の黒星街道の野を疾走していることをすごく自然なものとして感じている人が多いのではなかろうか。

パソコン商戦もそう。このほど各社の'89年度販売実績と'90年度計画がまとまったが、トップメーカーであるNECの販売台数は'89年度ついに100万台に到達。'90年度は120万台が目標と、もうとどまるところを知らない。販売金額も'90年度にはとうとう5000億円を突破するというからものすごい勢いだ。

2位グループ（エプソン、東芝、富士通、IBM）以下のメーカーとの差はますます開いてきたし、この2位グループとそれ以下との格差もこれまた大きくなってきた。起死回生の下位メーカー連合といわれたAXパソコングループはいつしか存在自体が風前の灯に。AXとは、APPENDIX（つけたじ）の略語だったのかしら？ という嫌みすら思ってしまう有り様だ。

だけどこれ、すごく自然に感じてしまっていてそれがどこか心地いいのが不思議なところなのである。

コンピュータビジネス自体の大型化、寡占化という宿命にもよるところは大きい、もはやソードのようなベンチャーの出る余地は全くないといつてよかろう。

そういえば、「ベンチャービジネス」という言葉が新聞紙上をにぎわせること自体がめっきり減った。内需拡大にともなう鉄鋼、建設、産業機械など従来型ビッグビジネスの活況が連日、報じられている。

数年前とは明らかに違う現象だ。こんなことはなかった。社会構造の変化にともない、微妙に風潮まで変わってきていることは指摘できる。すべて同一線上に積み重なっているストーリーとして考えると乱暴かもしれないが、おもしろい作業ではある。

だけど。

そうすると、安定社会というレールの上を走るだけの時代に逆戻りしてしまうのではないか、という不安を感じてしまう。

塾、一流高校、一流大学から官庁や大手企業に入り、いい家柄のお嬢サマと結婚して子供をレールの上に乗せる。

一瞬ゾッとするが、ふと現実を見ると、すでにそうなっているのだ。

まるで'60年代高度成長期のデ・ジャ・ブーを見ているような気がするのだが、実際に「岩戸景気」をしのぐ好況を迎え、似た雰囲気各方面でいろいろと出はじめているという。

これで「階級」の概念が出てくると終わりなのだが、これまた実は出てきている。

土地暴騰にともなう資産格差という経済的階級制フレームワークは実は暗黙のうちに土地暴騰にともなう資産格差によって、出来上がってしまった。しかも東南アジアからの労働者の台頭によって、より鮮明になってきているのだ。

恐ろしき本命主導時代。

歴史が繰り返すとするならば、オイルショックに似た経済破綻が襲ってくるはずなのだが。

さて。

ついに登場！ X68000SUPER-HD

Tan Akihiko 丹 明彦



1990年X68000ラインアップの最上位機種、
チタンブラックのX68000SUPER-HDの出
荷が始まった。概要は5月号でお伝えして
いるが、このほど編集部へ届いた製品をも
とに試用レポートをお届けする。

どこが新しいのか

PROIIやEXPERTIIも1990年バージョ
ンである。基本的には従来機種の設計を踏
襲しつつ、SX-WINDOWを装備してなお
かつ値下げを行っている(偉い!)。これに
対し、SUPER-HDは、もちろん基本仕様こ
そ変更はないが、PROやEXPERTとは一
線を画した設計になっているといってもいい。

SUPER-HDの目玉は、なんとといっても、
SCSIを標準装備した

ことであろう。SCSIとはハードディスクや
光磁気ディスクなど大容量外部記憶装置ほ
かさまざまな周辺機器に関する世界共通規
格のひとつであり、今後外部記憶装置の仕
様はSCSI規格に統一されていくものと思
われる。

X68000の従来機種のハードディスクは
SCSIに近いものではあるが純正のSCSI規
格ではない。そのため、X68000にSCSI規
格対応のハードディスクや光磁気ディスク
をつなごうとすれば、SCSIインタフェース
を新たに接続する必要があった。

これに対し、SUPER-HDはSCSI規格の
80Mバイトハードディスクを内蔵している。
シークタイムも短く、クラス最高速の部類
に入る。というわけで、OSもこれまでのも

のはそのままでは使えない。といってもHum
an68k本体は基本的に同じで、新しくSCSI
用のデバイスドライバを登録するようにな
っている。

したがって、ハードディスク周辺に関し
てだけ従来機種との互換性がない。まあ、
ハードディスクは持ち歩いたりする性質の
ものではないので不都合はないだろう。ま
た、いうまでもないことだが、フロッピー
ディスクへのアクセス関係には一切変更が
ないので、市販ソフトが使えないのではな
いかといった心配も無用である。

大容量ハードディスクや光磁気ディス
クなど、今後メジャーになっていくであろう
外部記憶装置を自由に扱えるようにするた
めにはSCSI規格への移行が必要である。
今回の仕様変更は将来への布石なのだと見
ることができよう。

ハードとともにソフトも少し変わる

X68000シリーズの常として、ソフトウェ
アレベルでの互換性は新製品でも完全に保
たれている。ただSCSI対応になったため、
ハードディスクなどに直接関わりそうな部
分は少々変更されている。システムディス
クのラベルを見ても、「SCSI対応」の文字
が輝いている。変更になったソフトウェア
のうちで主なものは、

COMMAND.X

FORMAT.XやDISKCOPY.Xなど

X-BASIC

といったところである。システム自体を始
め、軒並みバージョン2.02、2.10というも
のに変わっている。

DRIVEコマンドを使うとBASICでのデ
イレクトリ表示時にフロッピーディスクが
チェックされるというバグも今回フィッ
クスされているようだ。

ほかにも、ちょっと大きなところとして、
ビジュアルシェル(VS.X)が付属してい
ないことが挙げられる。EXPERTIIやPROII
にはちゃんとあったことを考えると、ビ
ジュアルシェルがSCSI規格に対応できてい

ないということであろう。

おそらく開発側ではCOMMAND.Xなど
のようなバージョンアップは不要だと判断
したのである。というのも、SX-WINDO
Wがビジュアルシェルの機能を十分カバー
できているからである。だから困ることは
まずない。今後はSX-WINDOWが主流に
なっていくということでもあるのだろう。

さて、SCSIドライバのほかにも、デバ
イスドライバの中には多少バージョンアップ
されたものがある。タイムスタンプを見る
限りでは、

OPMDRV.X

FLOAT2.X

の2つが新しくなった模様である。こちら
は、SCSI規格対応とは関係ないバージョ
ンアップであろう。

もちろん、これまでの機種同様、WP.X
(標準ワープロ)も付属するが、残念ながら
ワープロまわりはバージョンアップされて
いない。

もちろんマニュアルだって変わる

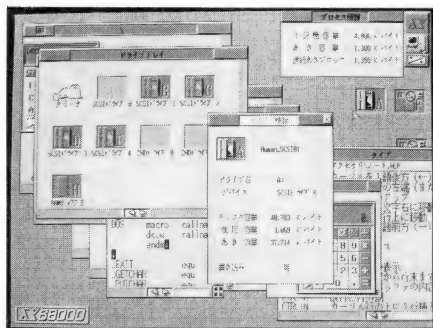
ビジュアルシェルがなくなり、SX-WIN
DOWが入った分、マニュアルも変わった。
内訳は、

- ・取扱説明書
 - ・Human68k ver2.0ユーザーズマニュアル
 - ・SX-WINDOWユーザーズマニュアル
 - ・X-BASICユーザーズリファレンス
 - ・日本語ワードプロセッサ・辞書ユーティ
リティユーザーズマニュアル
- となっている。

VS.Xがなくなったため取扱説明書の内
容が全面的に改められたほかは、SUPER-
HDだからといって、内容は特殊なもの
はない。PROIIやEXPERTIIにもほぼ同様
のマニュアルが付属している。

X68000の良さは外観にもある

1990年バージョンは、EXPERTII、PRO
II、そしてSUPER-HDともに、本体の「X
68000」のロゴがメタルエンブレムになって



SCSI デバイスサポート

いる。この外観にこだわるあたりがX68000魂を思わせてなかなかよい。

本体デザインは初期型からのマンハッタンシェイプ。キーボードもマウス・トラックボールも初期型からの伝統。ポップアップハンドルも基本である。本体色がチタンブラックというのは好き嫌いの分かるところだろう。

ディスプレイは例の音声多重（耳つきともいう）。チタンブラックのディスプレイはとりあえずこの1機種だけなのだが、高解像度の最上位機種なので、ディスプレイに関しては選択の余地は不要だろう。そうそう、21型ディスプレイもステレオになったのでよろしく。

広い広い80Mバイト

ハードディスクは買ったときのまっさらな状態では使えない。領域確保をしたあと、システムを入れて初めて機能するものである（物理フォーマットは初めからかけてある）。ハードディスクを使うことは、多少の予備知識が要求されることなのだ。ことにSUPER-HDの場合はハードディスクの容量が大きいの、システムの構築のしかたにユーザーのセンスが表れるといってもいいだろう。

ちょいと細かい話になるので恐縮だが、EXPERTやPROに付属のHuman68k（バージョン2.0）では、40Mバイトを超える容量の外部記憶は40Mバイトずつに分けて扱う必要があった。たとえば80Mバイトのハードディスクなら、40Mバイトのハードディスクが2台あるつもりで使うことになっていた。SUPER-HDのSCSIドライブにはそのような制限はなく、80Mバイトのハードディスクは80Mバイト1台である。もし望むなら、「80MバイトのAドライブ」を作ることにも可能になっている。

しかし、1台のハードディスクをひとつのドライブとしてしか使わない、そういう領域確保のしかたがあまり利口ではないことも事実である。では利口な領域確保とはなんだろう。1台のハードディスクに複数のドライブを同居させるのが賢いとされている。実例を挙げると、たとえば編集室のSUPER-HDの場合、システムドライブに40Mバイト、辞書のドライブに5Mバイト、作業用のドライブを2つ作ってそれぞれ10Mバイト取り、プログラムには15Mバイトの領域を確保している。

このように、用途別にドライブを分けておくのは大容量ディスクを使う際の常套手段である。ハードディスクは、使い込むに

つれて、膨大なファイルの情報が複雑に絡み合い、どうしても速度が少しずつ落ちてくるものだ。そんなドライブを整理するような場合でも、用途別のドライブを作っておくと実に便利である。とすると、この方法を採用する限り、80Mバイトだろうが40Mバイト×2だろうが使ううえではたいした差がないことになる（話に一貫性がない）。

それから、ぜひとも驚いてほしいのは領域確保の速さである。なんと数秒で済んでしまう。従来機種が領域確保に分単位の時間を要していたことを考えると驚異的ですからある。不良セクタチェックを省略したのである。最近のハードディスクは丈夫なので領域確保なんてたまにしかやらない作業だが、それでも速いのはいいことだ。

かくのごとく広くて速いハードディスクであるが、それより速い記憶装置がある。それはメモリである。SUPER-HDのメインメモリはEXPERTと同じで標準で2Mバイトだ。標準システムがSX-WINDOWだとすると、まだこれでも足りないような気もする。

最後に

初代X68000が発表された当時、僕は両面高密度フロッピーディスク（HDと略するのだ）をハードディスクと勘違いした経験が

ある。そのときは「この本体にハードディスクなんか入るわけじゃないか」と思ったものだった。ま、当時のハードディスクは20Mバイトでもけっこう大きかったし、X68000の本体からしてコンパクトなものだった（しかも縦置き）ことを考えると、ハードディスクの入る余地などないと考えたのも当然であろう。

ところが技術の進歩は実際たいしたもの、省スペース・高密度実装の結果、20Mバイトのハードディスクが載り、40Mバイトが載り、いまや80Mバイトである。

なんだかSUPER-HDの試用レポートといいながら、SCSIハードディスクの紹介に終始してしまったような気がするのだが、このことは、裏を返せば、X68000が頑固なまでにコンパチビリティを守り続けていることの証明でもある。初期型以来、どのモデルを使ってもまったく違和感を感じない（もちろんメインメモリの容量やハードディスクの有無で使用感が左右されることはあるが）。

値段そのものはPROやEXPERTに比べると少し高いが、中身の充実度を考えると十分に安い。増設用のSCSIボードも発売されるが、特に将来光磁気ディスクを導入したいとお考えの方はSUPER-HDを第一候補にあげるのが賢明だと思う。

新しいSX-WINDOW

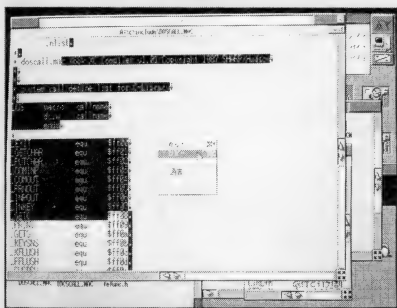
X68000SUPER-HDの発表はSX-WINDOWの正式な発表でもある。これまでもEXPERT II、PRO IIにはSX-WINDOWが付属していたが、それ以前のX68000ユーザー用には近日発売予定のままで発売は開始されていなかった。今回、新しいバージョンがリリースされ、別売りバージョンの発売がいよいよ秒読み段階に入ったといえるだろう。

といっても、今回のSX-WINDOWでもファイル検索時のメッセージ表示が加わった程度しか目に見える違いはないが、内部では細かなバグフィックスや改良が行われていると思われる（当然？）。文字表示速度なども若干改善されているようだ。

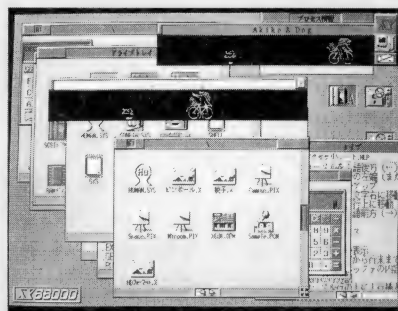
システム以外の部分での大きな変更はウィンドウエディタ、ノート、Xに見られる。吉田幸一氏がコメントを差し控えた前バージョンと比

べて、ウィンドウがカーソルと連動する、カット&ペースト機能が使えなどの改良がされている。なぜか、コントロールキーのキー操作がバツファに溜まるという改悪点も見られるのが残念だ。そのほか、小さなところでは、暁子さんの自転車に乗っているコンピュータがMacintosh（？）からX68000に変更されている。

SX-WINDOW用のマニュアルは操作法と付属ファイルの解説に終始しており、ウィンドウ上でのアプリケーション開発用資料、および開発システムの公表はまだしばらく先になる模様。早くウィンドウ上のアプリケーションを開発したいという方にはもうしばらくおあずけだ。市販のアクセサリにも大いに期待したいところだ。（S.N.）



カット＆ペーストもできる



新しい暁子さん

NEW PRODUCTS

情報ツール「All in Note」 AX286N-H2 シャープ



AX286N-H2

シャープはノート型のAXパソコン「AX286N-H2」を発売した。情報ツール「All in Note」と呼ばれるこのマシンには簡単に使いこなせるソフト「Business Mate」と、アプリケーションソフトの組み込みと起動が容易に行える「SPシステム」が標準装備され初心者にも使いやすくなっている。また、20Mバイトの容量を持つ2.5インチハードディスクを採用し、本体は厚さ34mm、重量2kgと小型かつ軽量である。

表示は冷陰極管バックライト（サイドライト方式）付きで鮮明な白黒表示のトリプルスーパーツイスト液晶ディスプレイにより鮮明で見やすい白黒表示（8階調）を実現。また、画面もAX仕様の640×480ドットと従来のノート型パソコンよりもひと回り大きくなっている。

さらに、オプションは2400bpsクラス5のモデムボード、CRTインタフェイス、増設RS-232Cなどの中から用途に合わせて内蔵でき、メモリも内部に最大2Mバイト（トータルで3Mバイト）増設できるので拡張しても携帯性を失うことがない。別売の拡張ユニット（本体と一体化できる）を使えばより豊富な拡張性が得られる。バッテリー駆動時間は約1.7時間で、一体型外部バッテリーを使えば5時間にできる。

価格は税別で398,000円。

〈問い合わせ先〉

シャープ(株) ☎06(621)1221, 03(260)1161

UNIXオフィスプロセッサ OA-120 シャープ

シャープはモトローラ製のMC68030を搭載したUNIXオフィスプロセッサ「OA-120」を発売した。このワークステーションは従来機種「OA-110WS」の2倍以上の性能を実現しており、OSには「UNIXシステムVリリース3.0」を採用している。また、高性能32ビットUNIXワークステーションながら、幅425mm×奥行き420mm×高さ120mmのコンパクトサイズを実現している。

「OA-120」ではハードディスクに加え150Mバイトのカセット磁気テープ装置が接続でき容易に大容量バックアップが行える。ほかにも液晶センサーパネルや、最大16台のPOSターミナルが接続可能である。

内蔵ハードディスクが40Mバイトと80Mバイトのものの2タイプがあり、価格はそれぞれ1,495,000円、1,795,000円(税別)となっている。

〈問い合わせ先〉

シャープ(株) ☎06(621)1221, 03(260)1161



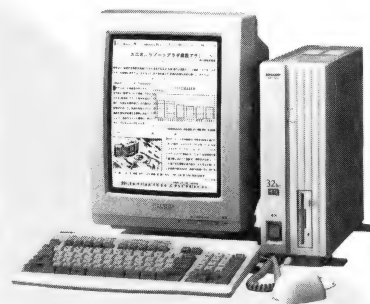
OA-120

書院シリーズ&ニュー書院 WD-A620/A720/7000 シャープ

シャープはパーソナルワープロ初のデュアルフォント&アウトラインフォントを内蔵した「書院」シリーズの新機種WD-A620/



WD-A720



WD-7000

720と、32ビットCPU搭載の「ニュー書院」WD-7000を発売した。

「書院」シリーズの新機種はパーソナルワープロとしては初の明朝体とゴシック体をROMで標準装備したデュアルフォントとアウトラインフォント機能、パーソナルDTP機能などの搭載により高度なレイアウト文書なども表現力豊かに美しく仕上げる事ができるようになった。プリンタは毎秒52字の高速印字で、高密度印刷のできる56/52/48ドットのものを採用している。また、辞書にはAI-V3辞書を搭載しており口語文変換などの変換効率を高めている。そのほかの機能としては電子手帳とデータの共有ができる電子手帳通信機能、個人データの管理に便利なパーソナルデータ管理機能などがある。さらに、WD-A720は上記に加えて3.5インチフロッピーディスクドライブを2基搭載（WD-A620は1基）しており、通信ソフトやMS-DOSコンバータなども標準で備えている。価格はそれぞれ145,000円と178,000円（税別）。

書院シリーズの最上位機種である「WD-7000」は32ビットCPUを搭載し処理速度を大

幅に向上している。「WD-7000」は40Mバイトハードディスク、そしてA4フルページ表示(24ドット)が可能な高解像度15インチ縦型CRTを内蔵、さらにA3サイズ対応の高解像度400DPIレーザープリンタを用意するなど高性能なビジネス用日本語ワードプロセッサとなっている。

また、矩形単位の移動・複写・切り取り・貼り付け機能による編集作業の効率アップ、アウトラインフォントおよび正楷書体標準装備による表現力の向上の実現なども果たしている。価格(税別)は標準システム(本体、CRT、キーボード、マウス含む)が1,208,000円、レーザープリンタシステムが2,150,000円。

<問い合わせ先>

シャープ(株) ☎06(621)1221, 03(260)1161

CCITT規格V.42bis, 38400bps Multi modem V32L コア

コアは「Multi modemシリーズ」の新機種を発売した。「Multi modem V32L」はCCITT(国際電信電話諮問委員会)がモデム用データ圧縮方式のプロトコルとして採用した「V.42bis」を搭載している。CCITT規格V.42bis機能を搭載したモデムとしては日本で初めての発売となる。機能は一般回線用モデムとしてV.42エラー訂正、セキュリティコールバック、リモートコンフィギュレーションなどを装備し、V.42bisデータ圧縮機能により38400bpsの超高速を実現している。また、2線および4線専用回線にも利用でき4線専用回線に対する一般回線による自動バックアップおよび自動復帰機能を持っている。価格は298,000円。

<問い合わせ先>

(株)コア ☎045(441)8611

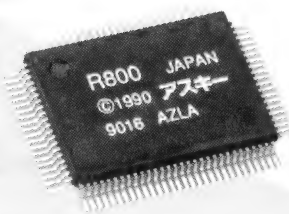


Multi modem V32L

Z80完全上位互換16ビットCPU R800 アスキー

アスキーはザイログ社のZ80と完全上位互換性をもつ高速16ビットマイクロプロセッサ「R800」を開発した。「R800」は同一

R800



クロックのZ80と比較して3.6倍の高速動作を実現している。また、メモリ空間はメモリ拡張回路により16Mバイトまで拡張可能である。そのほかの特徴としては、高速な16ビット掛け算命令のサポート、2チャンネルのDMAコントローラ内蔵、7階層の優先順位をもつ強力な割り込み機能などがあげられる。

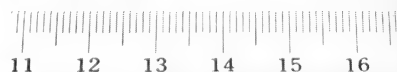
この「R800」はZ80互換マイクロプロセッサとして世界最高速の性能を実現している。このように高速化できた背景には、16ビットALUバスの採用、RISC CPUなどで用いられるパイプライン処理により1命令あたり平均1.3CPUクロックに実行速度が上がったこと、28MHzの高速クロック動作(CPUクロックは4分の1)の実現などがある。

そして、表面実装に最適な100ピンQFPに搭載することで従来の40ピンDIPと比べ体積比で3分の1以下の小ささを実現し、CMOSプロセスを使うことで低消費電力となっている。

<問い合わせ先>

(株)アスキー ☎03(797)6506

レーザーピックアップ駆動回路のIC化 IR3C10/IR3C11 シャープ



シャープは光ディスク用レーザードライバ「IR3C10」とレーザーコントローラ「IR3C11」を開発した。光ディスクは記憶容量の大きさなどから次世代の記憶メディアとして期待されているが、アクセスタイムの点ではまだ技術的には成熟しておらず、磁気ディスクに比べ劣っているといわれている。この要因のひとつとしてレーザーピックアップの大きさと、ピックアップスライドモーターの重量が考えられ、ピックアップ内に搭載される半導体レーザードライバ部のIC化が望

まれていた。IR3C10およびIR3C11はこれに応えるものである。レーザーピックアップ部の小型・軽量化を考慮してコントロール部はピックアップ外に配置することを想定したため、あえて2チップ構成になっている。

<問い合わせ先>

シャープ(株) ☎06(621)1221, 03(260)1161

OS-9/X68000用

SrcDbg V2.0 マイクロウェア・システムズ

マイクロウェア・システムズはOS-9/X68000上で動作するマイクロウェアCコンパイラ用のソースレベルデバッガSrcDbgを発売した。これを使用してCプログラムをインタラクティブに実行することにより、プログラムの開発時間を短縮することができる。また、アセンブリ言語レベルでもOS-9付属のDEBUGコマンドとほぼ同様にデバッグできる。パッケージ内容は、

ソースレベルデバッガ SrcDbg

ヘルプファイル

練習用サンプルプログラム Where.c

SrcDbgユーザーズマニュアル

となっていて、価格は39,800円。

<問い合わせ先>

マイクロウェア・システムズ(株)

☎03(257)9000

INFORMATION

シャープ ワールドサッカー'90 in 東京



オールドリッジ
(レアル・ソシエダ)

レナト(フラメンゴ)

来たる8月6日(月)に東京ドームで「シャープ ワールドサッカー'90」が開催される。この大会は昨年から行われているものであるが、本年度はブラジルの名門チーム「CRフラメンゴ」とスペインの強豪「レアル・ソシエダ」が来日して対戦する。チケットはすでに発売中。入場料金は自由席2,000円(大人)、1,000円(小・中学生)から。

<問い合わせ先>

シャープ(株) ☎06(621)1221, 03(260)1161

FILES Oh!X

このインデックスは、タイトル、注記——
筆者名、誌名、月号、ページで構成されて
います。うっとうしい梅雨も終わりです。
さあ、夏休みに向けて楽しい計画を立てま
しょう。補習のある人は別ですけど……。

一般

▶ ネットワーカー・ホリック第20回

アイドル情報が自慢のフジサンケイ系ネット「EYE-
NET」と、集英社のコミック雑誌編集部直営のネット「BJ
ネット」を紹介。PDSはPC-9801のゲーム「クアトロ」「タ
ムタム」を紹介。——編集部、LOGIN、9・10号、214-215
pp.

▶ アルゴリズムを見切ったぞ!!

ブロック崩しのボールの動きを処理するアルゴリズム
を例に、プログラムの組み方を解説している。——編集
部、テクノポリス、6月号、100-104pp.

▶ ディスクファイルのトラブル 注意点と復活対策

パソコンが広く使われている現在、そのデータの保守
管理はますます重要になっている。ここではフロッピー
をめぐる物理的・論理的なトラブルの内容とその対処法
について考える。——佐田守弘、マイコン、6月号、138-
151pp.

▶ MICOM COMEON GOODS

今月は、最近人気のコンピュータミュージック機器を
まとめて紹介する。モニタスピーカやMIDI用ソフトウ
ェアなどなど。——編集部、マイコン、6月号、158-163
pp.

▶ シャープ 春の見・体・験フェア

4月14・15日新宿NSビルで行われたシャープのフェア
の模様をレポートする。——編集部、マイコン、6月
号、166p.

▶ シミュレーションウォーゲーム研究

XI turbo 用にも発売が決定している「大航海時代」につ
いて、光栄のシブサワ・コウ氏が解説。——シブサワ・
コウ、マイコン、6月号、188-192pp.

▶ LET'S PROGRAM

今月の宿題は「階乗の計算」。オーバーフローを起こさ
ぬようテクニックを凝らしたサンプルがX-BASIC、アセ
ンブラ、XI用 BASIC で発表されている。——藤本健、マ
イコン、6月号、231-240pp.

▶ ここまで来た DTP の世界

日本でもサムシンググッド社から発売になった「アル
ダス・ページメーカー」。ソフトの概要と併せて DTP パブ
リケーションとはどういうものかを詳細に追う。——稲
澤薫、マイコン、6月号、257-271pp.

▶ パソコン横丁

先月に引き続いて、文字列検索のアルゴリズムにつ
いて考えるおしゃべり風アルゴリズム講座。今月はとっ
ても速い BM 法を扱う。——やまさん、マイコン、6月号、
300-304pp.

▶ マイクロマウス マッピージュニア

ナムコから迷路を解くロボット「マッピージュニア」

が発売になった。その構造、パソコンをつないでの遊び
方についての解説記事。——時国修、マイコン、6月号、
321-327pp.

▶ 実践ハード入門

今月は磁気センサの製作。ホール素子を使って磁界の
強さを測る。ミニ知識「電磁誘導」付き。——石川至知、
マイコン、6月号、347-351pp.

▶ ハードディスクであなただが変わる

最近普及の目ざましいハードディスクの入門記事。ハ
ードディスクとは何かから始めて、その利点、種類、
購入などについて述べる。——吉沢正敏、I/O、6月号、
82-88pp.

▶ 東側ハイテク事情

東欧改革以後、西側のハイテク企業が続々と市場に参
入している。その影響と未来についてのレポート。——
ジョン・ヒルカーク、I/O、6月号、220-221pp.

MZ シリーズ

▶ 誌上公開質問状

MZ-80K/MZ-2000 のカセットの修理を受け付けてくれ
るところは？ MZ-2500 用のマシン語の本は？ などの
質問に答えている。——PEGASUS、マイコン BASIC Ma
gazine、6月号、90p.

MZ-1500 (MZ-5Z001 BASIC)

▶ J 君の奇妙な冒険

柱をのぼっていく波紋作業ゲーム、あぶくをよける仙
道ゲームの2つがミックス。——FROG、マイコン BASIC
Magazine、6月号、125-126pp.

MZ-2500 (MZ25-BASIC)

▶ がんばれ! 孫五郎

雲に乗ってぼんぼこ山に咲くガビタンの花をすべで摘
む、パズルゲーム。——謎のバズル大好きおじさん、マ
イコン BASIC Magazine、6月号、127-128pp.

X1/turbo/Z

X1 シリーズ

▶ 画像圧縮プログラム

XI 用の画像圧縮プログラムを BASIC で。データ圧縮に
ついての参考にもなる。——山本克稔、マイコン BASIC
Magazine、6月号、73-74pp.

▶ 7 殺し

7 並べの変形版カードゲーム。——ポケタ玉井、マイ
コン BASIC Magazine、6月号、155-157pp.

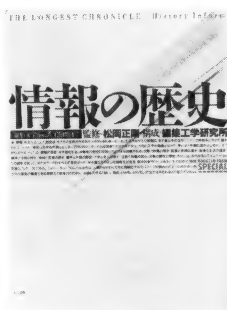
▶ FRONTIER

20 年間でどれだけ街を発展させられるかを競う。都市
開発ゲーム。——バル、マイコン BASIC Magazine、6月
号、158-160pp.

参考文献

I/O エ学社
ASCII アスキー
コンプティーク 角川書店
テクノポリス 徳間書店
POPCOM 小学館
マイコン 電波新聞社
マイコン BASIC Magazine 電波新聞社
LOGIN アスキー

新刊書案内



デザインに戸田ツトムをフィーチャーした NTT
出版の BOOKS IN ▶ FORUM シリーズは情報・コ
ンピュータ関係の怪しげで面白そうな本を出して
くれる。今回の「情報の歴史」もまた格別である。
枕許に置いて、いつでも手に取って眺めていたい。
この本はオールカラー写真ナシのただの年表で
ある。タイトルにある「情報」は「人間が獲得し改
良しつづけてきたコミュニケーションのためのす
べての手段および内容。」(「関係の発見のために」
より) のことである。そういったことであるから、
紀元前 7000 万年前から 1990 年までの壮大な年表だ。
むろん、1990 年は白紙。教科書の副教材にあるよ

うな退屈な年表とは思ってはいけな。独自の切り
口で、文字情報だけでなくながら平板でない面白
い年表。これを作成する労力と知力を考えるとど
っとするくらいだ。この本は歴史を調べるため
にあるのではなく、眺めながらいろんな発見をする
ためにある。「情報」をおいけることは人の「文
化」を見ることであり、「コミュニケーション」の
歴史をつかむことである。高いけど、値段分の価
値はあるぞ。(K)
情報の歴史 松岡正剛監修 編集工学研究所構成
NTT 出版
☎03(435)1212 A4判 433ページ 4,800円

書籍の価格は消費税込みです

X1+FM 音源ボード(要 NEW FM 音源ドライバ)

▶ THE SUPER 忍 LONG DISTANCE

メガドライブ版ザ・スーパー忍のミュージックプログラム。——木村直之, マイコン BASIC Magazine, 6月号, 187-188pp.

X1turbo シリーズ

▶ NEW SOFT

セレクトッドソーサリアンを紹介。——編集部, LOG IN, 9・10号, 24p.

▶ How To Win

セレクトッドソーサリアンを紹介。——編集部, コンピューター, 6月号, 132-135pp.

▶ 先取りおすすめゲーム

6月発売予定の大航海時代を紹介。——編集部, テクノポリス, 6月号, 8-11pp.

▶ 月刊ソーサリアンニュース

セレクトッドソーサリアンの妖夢幻伝説, 闇に消えた女神を紹介。——編集部, テクノポリス, 6月号, 26-27pp.

▶ AIR FORCE

弾を撃ちあい, 相手のヘリを落とす。2人用。——山田仁, マイコン BASIC Magazine, 6月号, 161-162pp.

X68000

▶ 激突!! 対戦ゲームが新しい

近頃流行りの2人対戦可能なゲームで遊ぶ。X68000用は対戦ボピュラスを紹介。——編集部, LOGIN, 9・10号, 134-137pp.

▶ X68000新聞

新着ゲームの「ウルティマV」, 「ブレード オブ ザ グレートエレメンツ」, アイデアプロセッサハイパーテキストワープロ「Hyperword」, バージョンアップしたサイクロン「サイクロンα68」の紹介。SX-WINDOW続報。——編集部, LOGIN, 9・10号, 148-153pp.

▶ 最新ゲーム徹底解剖!!

ボピュラス紹介の第3回と, グラナダ。対戦ボピュラスと, 自分で面をエディットするカスタムモードを解説。グラナダはステージ1から3を攻略——編集部, LOGIN, 9・10号, 184-187pp.

▶ X68000SPIRITS

ボピュラス, FAR SIDE MOON, サークの紹介と, プロテニスワールドコート, ルーンワースの移植情報。——編集部, コンピューター, 6月号, 240-241pp.

▶ 先取りおすすめゲーム

7月下旬発売予定のウルティマVを紹介。——編集部, テクノポリス, 6月号, 12-15pp.

▶ GAMING WORLD

新作 & 移植情報。戦略シミュレーションゲーム「JOS

HUA」, ファンタジーアクション「バルーサの復讐」を紹介。——編集部, テクノポリス, 6月号, 30-32pp.

▶ WE ARE THE X68000 WORLD

新着ゲームと発売予定のゲーム「ラグーン」, 「ダウンタウン熱血物語」, 「スライミヤ」, 「バルーサの復讐」, 「コース」, 「FAR SIDE MOON」を紹介。——編集部, POPCOM, 6月号, 80-83pp.

▶ 誌上公開質問状

SYS ファイルの転送のやり方, PATH 指定の方法, マシン語を始めるのに必要なソフトは? などの質問に答えている。——多田太郎, マイコン BASIC Magazine, 6月号, 89-90pp.

▶ HOT INFORMATION

X68000ハイパーテキストワープロ「Hyperword」を紹介。——編集部, マイコン BASIC Magazine, 6月号, 91p.

▶ SORCERESS

一人前の魔女になるために修業の旅に出たあなた。ほうきに乘って魔物を倒そう。——林純一, マイコン BASIC Magazine, 6月号, 163-164pp.

▶ DROP BLOCK

テトリス+オセロ!? 対戦パズルゲーム。——永井崇博, マイコン BASIC Magazine, 6月号, 165-167pp.

▶ ゲームレビュー

アルシソフトの「ナイトアームズ」を取り上げる。斬新な試みに高い評価を与えているが, 熟成度はイマイチか?——あゆかわさつみ, マイコン, 6月号, 198-199pp.

▶ X68000マシン語入門

グラフィック関係のシリーズ第2回。グラフィックを加工する便利なプログラムを4本収録。色分布の表示や平均化など。——高橋雄一, マイコン, 6月号, 338-346pp.

▶ なんでもQ & A

「辞書ディスクをRAMディスクに「DISKCOPY」しようとしたらできなかった。なぜ?」「X68000で使えるカラーインクジェットプリンタは?」などの質問にシャープの人がバッチリ答えちゃうぞ。——編集部, マイコン, 6月号, 408-409pp.

▶ SOFTBOX

シャープから発売になった「Hyperword」。その概要を紹介する。——L&M, I/O, 6月号, 126-127pp.

▶ Disk Optimizer

使いこんで不連続クラスタばかりになってしまったディスクを整理するユーティリティ。作業用ディスクが不要, HDにも対応していることなどがウリ。——高城英誌, I/O, 6月号, 144-154pp.

▶ 常駐 HELP

いつでもドキュメントファイルが呼び出せるというユ

ーティリティだ。ASCII コードの表示・拡張ファンクションキーなども使えるぞ。——ずん, I/O, 6月号, 155-159pp.

▶ GAME BOX

メタルサイトとワンダラーズ・フロム・イースのレビュー。——市原昌文・吉沢正敏, I/O, 6月号, 166-167pp.

▶ YE

以前掲載されたフルスクリーンエディタの最終バージョン。細かい修正とバグ・フィックスが行われている。——井本祐司, I/O, 6月号, 177-187pp.

▶ 画面セーブプログラム

割り込み型の画面セーブプログラム。メニューでのオプション選択もできるので, フラクタル作成中の画面退避も可能。——原道宏, I/O, 6月号, 200-205pp.

▶ 白黒ビデオ画像入力装置の試作・実験

イメージユニットは高い! という人のために, イメージユニット用インタフェイスを使って自作の画像入力装置をつなごうというわけ。——ベリカン君, I/O, 6月号, 250-255pp.

▶ AV プログラム講座

X68000を例にして, グラフィックイメージの拡大・縮小・回転と, ラスタスクロールのアルゴリズム, そしてそれらの応用について解説する。——中山進・仲田津宏, ASCII, 6月号, 249-256pp.

▶ WSEL

OS-9/X68000用のファイルセレクト。ただしパーソナルウィンドウシステムが必要。——野出久司, ASCII, 6月号, 328-329pp.

▶ AV STRASSE

EXPERT II/PRO II から標準添付された SX-WINDOW Ver.1.0と, Console/GraphicsIOCS Ver.1.0の試用レポート。——仲田津宏, ASCII, 6月号, 345-347pp.

ポケコン

PC-E500

▶ GHOST CATCH

2~4人用ゲーム。ひとりがゴーストになり, 残りのメンバーがそれを追う。——せとけん, マイコン BASIC Magazine, 6月号, 173p.

▶ だいちゃんの日記 ~バレー編~

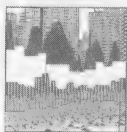
対コンピュータ・バレーボールゲーム。相手は4人から選べる。——神谷栄治, マイコン BASIC Magazine, 6月号, 174p.

▶ ポケットコンピュータ活用研究

先月扱った表計算プログラムを使って, PC-9801の「Lotus 1-2-3」とデータをやりとりする。データファイルの形式や, データ交換の手順など。——塚田洋一, マイコン, 6月号, 328-334pp.

作品としてのプログラム

黒川利明著



岩波書店

作品としてのプログラム

プログラミングやプログラマの実態, 心得, 考え方, 知識。著者の基本は, プログラムは人間が作る人間臭いもの, ということだ。読み物としてもいいが, ちゃんと読めばプログラミングや計算機の勉強にもなる。日本語とプログラム, モデル化からオブジェクト指向, 人工知能まで幅広く鋭く鋭く語られている。プログラミングを学ぶことは言語修得やテクニック修得では決してない。長生きできるプログラマの条件付。(K)

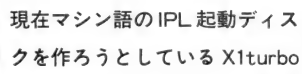
黒川利明著 岩波書店 ☎03(265)4111 B6判
199ページ 1600円



バックランドでつかまえて

東京トンガリキッズをご存じか? あれはトンガリキッズの物語の断片集だった。これもそう。ゲーセンをめぐるゲームキッズの物語の連作編集集。それがどうだ。面白いではないか。ゲーセンをひとつの文化として育った私たちには, イタリア製のおしゃれなアイテムを並べるよりクレイジークライマーをめぐる物語が妙にしくりくる。そして, これだけゲームと人間の関係, ゲーマーの気持ちを正しく描いた物語はほかにはない。(K)

田尻智著 JICC 出版局 ☎03(234)4621
B6判変形 199ページ 900円



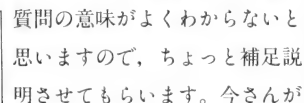
を参考にしてできた！ と思ったのですが、やはりダメでした。

ますが、僕のIPL用ディスクの考えは、

$$140 = 100(\text{FCB}) - 2, \quad 140(\text{FAT}) =$$

いったんここが間違っているのかもしれない。

します。青森県 今 隆範



送ってくれたテキストには、TEST.Sysと

ながらといって真向の中にも出てくる。

[illegible]

より、レーニンの内容を確認してみるこ

IFL から起動することの出来るプログラム

ゲストI Bを死にたい。これは7回

● 2013 年 12 月 1 日起, 凡在《劳动合同法》施行前已订立劳动合同, 且该合同未到期, 但用人单位未依法为劳动者缴纳社会保险费的, 劳动者可以依照《劳动合同法》第三十八条的规定解除劳动合同, 用人单位应当支付经济补偿。

これはノノイルの種類を表すもので、第

尤に遅れて 31 日下日はクアイルの倍額

 $5 \times 10^4 - 500$

を言さなければいけません。

たいしいのマンシヨ語ノオイルはレコード

このことを頭に入れて、今さんがITL起

ところが、この値を逆にすれば万事解決

A

```
:C010=73 20 00 38 00 00 00 00 00 00 00 00 00 00 20 00 /в .8.....
```

B

```
:C010=6A 20 00 02 00 C0 00 C0 89 52 15 23 53 00 05 00 /j ...タ.外 R.#S...
```

C

```
:C010=73 20 00 38 00 00 00 00 00 00 00 00 00 00 00 02 00 /a .8.....
```

D

:C010=73 20 38 00 00 80 00 80 00 00 00 00 00 00 20 00 /s 8.....

23, 24バイトに実行アドレスを、いずれも下位, 上位バイトの順で表すことになっていますので、このファイルは格納先頭アドレス, 実行アドレスともに0番地となっていることがわかります。はたして0番地から実行されるものであってよいのか疑問だったのですが、逆アセンブルしてみるとリロケータブルファイルであることがわかったのです。「リロケータブルファイル」とは、どんなアドレスにプログラムが置かれていても実行することのできるファイルのことをいいます。

となると、どこの具合が悪いのでしょうか。それはこのプログラムがBIOS ROMを使用していることに起因しています。周知のとおりBIOS ROMを使うプログラムは8000h以降に置かなくてはならないことになっていますから、格納先頭アドレス, 実行アドレスを8000hに変更して起動すれば…やっとうまくいきました。これでFATが破壊されているという疑いは消えたことになります。結局リスト1-Dが正しいレコード0のかたちです。



X68000とCZ-8PC4で複数文字にわたる外字を印字したいのですが、WP.Xでは48ドット印字ができませんし、ほかの方法では昨年の7月号に掲載された「PRNDRV0.SYS」を用いてもつながってくれません。なにかよい方法はないのでしょうか。

それからWP.Xの環境設定の部分などをして文字をビットイメージ出力でなくコード出力にして、48ドット文字を印字するようにできないでしょうか。

兵庫県 中田 勝啓



WP.Xを起動して画面右上の「その他」をクリックしてみてください。プルダウンメニューの書式設定を選択すると、書式設定のウィンドウが開いたでしょう。その中の文字間に注目してください。文字間には密着, 狭い, 普通, 広い, と4種類ありますが、これを密着にすれば複数文字にわたるロゴなどを定義した外字をくっつけて印字することができます。

図1

Q X68000とCZ-8PC4で複数文字にわたる外字を印字したいのですが、WP.Xでは48ドット印字ができませんし、ほかの方法では昨年7月号に掲載された「PRNDRV0.SYS」を用いてもつながってくれません。なにかよい方法はないのでしょうか。

それからWP.Xの環境設定の部分などをして文字をビットイメージ出力でなくコード出力にして、48ドット文字を印字するようにできないでしょうか。

兵庫県 中田 勝啓

図2

Q X68000とCZ-8PC4で複数文字にわたる外字を印字したいのですが、WP.Xでは48ドット印字ができませんし、ほかの方法では昨年7月号に掲載された「PRNDRV0.SYS」を用いてもつながってくれません。なにかよい方法はないのでしょうか。

それからWP.Xの環境設定の部分などをして文字をビットイメージ出力でなくコード出力にして、48ドット文字を印字するようにできないでしょうか。

兵庫県 中田 勝啓

ところが、この設定は困ったことに文書全体にわたって効果が出てしまうので、長い文書を印字するときは文字がくっついて読みづらいことになるでしょう。そのために密着部分だけあとで印字するとか、ユーザー自身が工夫しなくてはとても使いものにならないでしょう。PC-9801用のワープロでは文字単位に密着指定できて当たり前となっています。

図1が通常の出力、図2が密着指定をしたときの出力です。これでは文字が詰まりすぎなので、文字間に半角スペースを挟んで調整したものが図3です。これは、ファイル出力した文書をエディタで読み込み、

ESC・@

カーソル右

スペースキー

UNDO

(しばらく待つ)

ESC

のように操作します。

さらに、手作業で調整すれば図4のようになります。

それからWP.Xの出力をコード印字にすることができるのか、という質問ですがこ

図3

Q X68000とCZ-8PC4で複数文字にわたる外字を印字したいのですが、WP.Xでは48ドット印字ができませんし、ほかの方法では昨年7月号に掲載された「PRNDRV0.SYS」を用いてもつながってくれません。なにかよい方法はないのでしょうか。

それからWP.Xの環境設定の部分などをして文字をビットイメージ出力でなくコード出力にして、48ドット文字を印字するようにできないでしょうか。

兵庫県 中田 勝啓

図4

Q X68000とCZ-8PC4で複数文字にわたる外字を印字したいのですが、WP.Xでは48ドット印字ができませんし、ほかの方法では昨年7月号に掲載された「PRNDRV0.SYS」を用いてもつながってくれません。なにかよい方法はないのでしょうか。

それからWP.Xの環境設定の部分などをして文字をビットイメージ出力でなくコード出力にして、48ドット文字を印字するようにできないでしょうか。

兵庫県 中田 勝啓

れは無理です。どうしてもコード印字をしたいのなら、4月号の質問箱でとりあげたようにWP.Xの印刷機能を使わずに、自作のプリントアウトプログラムを使うようにすればいいでしょう。確か、以前電脳倶楽部にも48ピンプリンタ対応の印刷プログラムが載っていたと思います。(影山 裕昭)

質問にお答えします

日ごろ疑問に思っていること、どんなことでも結構です。どんどんお便りください。難問、奇問、編集室が総力を上げてお答えいたします。ただし、お寄せいただいているものの中には、マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限、マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名、システム構成、必要なら図も入れてこと細かに書いてください。また、返信用切手同封の質問をよく受けますが、原則として、質問には本誌上でお答えすることになっていますのでご了承ください。なお、質問の内容について、直接問い合わせることもありますので、電話番号も明記してくださいね。宛先：〒102 東京都千代田区

九段南2-3-26井関ビル
株日本ソフトバンク出版部
「Oh!X質問箱」係

FROM READERS TO THE EDITOR

この本をみなさんが手にするのは梅雨真
つ只中の頃と思います。いま外では雨が
降っているでしょうか。雨ってきれいな

人が多いと思うんですが、まあまあ好き
なんですよね。なんか気分が鬱になりま
すからね。変人なのかな? (A)

◆来月(6月号)はフロッピー付きかあ……。
もしFLOAT2+.XやOPMD.Xが収録されてたら
ショックだなあ。だって春休みにアルバイトの
合間にちびちびとがんばってうちこんだんも
ん……。ううっ。 石間 崇(19) 京都府
入っていましたね。かわいそうに……。うう
っ。

◆ディスクが付くからって値段が上がるのはコ
ンピュータ雑誌多しといえどOh!Xぐらいじゃな
いか? 中身がよければ許す!

北条 章(37) 東京都
中身はともかく、普通に原稿料を払えばあ
れくらいにはなるんですよ。

◆SUPER-HDは本屋さんでもらえるのですか。
みんなが知ったらきっと本屋さんがつぶれるぐ
らいの人が来そうだから、予約しといたほうが
いいのだろうか。ありがとうございます。

梶田 真二(16) 奈良県
だから、ちがうって……。

◆6月号の特集「Oh!Xも8周年だからたった220
円ぼっち増えただけでディスクを付けてやる
ぜ! ありがたく思え」に期待させていただき
ます。 小堀 昌宏(21) 兵庫県
そんなにいかたしなくても……。

◆6月号はすごそうですね。悪魔のツールって
なんなのでしょう。気になります。ところで、
読者から投稿されたプログラムの入ったディ
スクはどうなっているんでしょう。ふと気になっ
てしまった。そうとうの数になっていると思う
んですが。ま、いいけどさ。

奈良 雅雄(16) 栃木県
読者から投稿されたディスクですか。えっ
と、フォーマットして使っています……。ウ
ソですよ。

◆こんばんは。いまは6時21分です。編集室の
人たちはどのフロッピーを使っていますか?
また、どのフロッピーがいちばん良いと思
いますか。 安元 英彰(21) 兵庫県
だから、読者から送られてきたフロッピー
ディスクをフォーマットして……。(みなさ

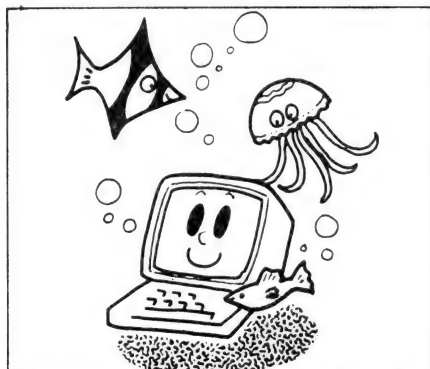
ん本当にウソですよ)。

◆がーん! 5月号 P.44 の発売中のソフトに
「ギルガメッシュ・ソーサリアン」が。もう、
売っていたとは。その翌日タケルへ走った私は
再び、がーん! まだ売ってねーちゃんか。
メーカーに電話してもらおう「メドが立たない、
6月中には……」だ。「ごめんなさいのコーナ
ー」行きだよ、コレは。

味野 真一(24) 岡山県
どうも開発の進行が思ったようにいかな
かったようで発売されなかったようですね。
どうも、ごめんなさい(ゲーム記事担当
者)。

◆ポピュラスは受験生の敵だ! 全国の受験生
を代表して僕がポピュラスを踏んづけてくしゃ
くしゃにします。今すぐ身柄を引き渡し
てください。 萩野 真一(17) 愛知県
とかいいながら、自分がハマってるん
ですよ。

◆懺悔。職業欄に「来年は大学生」とか「春に
は大学生」などと書いていましたが、桜散るど
ころではなく桜狂吹雪となり何も見えなくな
りました。ハガキに書いたことはぜ〜んぶウソ
となってしまいました。これらの罪は重く懲役



1年、罰金50万円の予備校の刑に処せられまし
た。来春、晴れて出所のあかつきには「合格し
たよ〜」とご挨拶したいと思います。

立花 真哉(19) 茨城県
浪人もなかなかいいものですが、あんま
り予備校生活が長くなりすぎて半名主にな
ってしまわないようにしましょう。

◆と一と一社会人になりました。思えば学生時
代はなんて自由だったのでしょうか。そう思って
毎朝学生を見るとやっぱりあの頃はよかったな
ァと思うけど、お金をカボツともらうと社会人
はこれだからやめられねー(?)と思います。

山本 裕治(18) 滋賀県
まさにそのとおりですよ。

◆とうとう高3になってしまいました。学校が
就職校なので働くことになるんです(もちろん
進学もできますが)。友達そのことを話すとや
れアセンブラを勉強してソフトハウスに就職す
るなどできもしないことを平気でいっていま
す。僕もどうなるかわからないけど、もしどう
しようもなくなったらOh!Xのスタッフに入れて
もらえませんか。 村松 智行(17) 静岡県
そうか、Oh!Xのスタッフってどうしようも
なくなった人間の集まりだったのか……。

◆大学行けませんでした。夢はかないません
でしたが私は一労働者となり一定の「金」が手
に入る。フフフ、X68000SUPER-HDを注文したいま
となつては6月が待ちどおしいぜ。C compiler
PRO-68Kを注文してこよっと。おっと、色はもち
ろんチタンブラックさ。

笹野 暢彦(19) 静岡県
人間どこに幸せが転がっているかわからな
いのですね。

◆4月から大学生、しかもひとり暮らし。思
いっきりX68000をやるぞ! と思っていたのに、
炊事・洗濯 etc. と結構忙しい。それに自分が家
計をあずかっていると、ソフトや周辺機器を買
うのもお金がもったいないという気持ちになっ
てしまう……。 松永 正弘(20) 京都府
う〜ん、大学生ではなくすっかりしっかり
ものの主婦と化していますね。適当にやっ
ていてもなんとかなるものですよ。

◆いま僕は理科の勉強をしています。あつ、先



◆丸藤 俊之 神奈川県
丸藤さんは毎月のように載っているのだから今回は遠
慮してもらおうと思ったのですが。う〜ん、これ
だけのものを送って来られると……。カッコイイ。

生に見つかってしまいました。先生の右手が僕のえりをつかみました。「うわいま先生の説教を受けています。あー、今度は左こぶしが上がりました。「1秒ごとにエキサイティング」もうだめです。みなさんさよならああああ……「ボカ」。

芝 哲也 (16) 広島県

みなさん、くれぐれも見つからないように読みましょう。

◆うちの学校の物理科学部員はPC-9801を運べといわれるといやがる。だが、個人で持ってきたX68000を運んでと頼まれると取りあいをして「うおおー、初めてキャリングハンドルにさわった一つ」と喜ぶ。藤田 真人 (16) 静岡県

X68000PROの場合はどうなるんだろう。

◆月2回になるとうれしい。

今村 雅 (17) 長野県

月2回になるとしんどい。

◆最近内容は濃いので全部読むのに2カ月かかる。武川 博 (20) 静岡県

ということは2カ月に1度しか買わずに済むということだ。じゃなくて、ちゃんと毎月買ってね。

◆僕のOh!Xが5冊になりました。並べると厚くなってなんかうれしい。

小椋 昌宏 (16) 三重県

この本が薄いということがいいたいのじょうか (深読みしすぎかな)。

◆Oh!Xを初めて買いました。こんなに中身の充実している本を読んだのは辞書以来です。これからは読み続けるのががんばってください。

柴田 知久 (17) 東京都

辞書のように中身はあるけどつまらないといいたいのだろうか…… (ちがうって)。

◆ようやくX68000を購入しました。Oh!Xも毎月買おうと思います。THE COMPUTERも毎月買っています。C MAGAZINEもときどき買っています。Oh!PCはたまに買っています。月刊情報処理試験、先日たまたま買いました。Oh!FMは以前買ってました。プレゼントください。

有山 州一 (31) 福岡県

まだまだ甘いですね。パソコン・マガジンやBEEP! メガドライブ、それにPC WEEKも買ってくれないと。

◆みっ、見てしまった。ついに生で見てしまった(変なものじゃないぞ)。学校で後ろの席のヤツが、あわわわ……(おどろき)。そいつは手もふれずスプーンを曲げてしまったのだ! 「いますぐにできる?」と聞くとなんのためらいもなしに「できるよ」。たまたま弁当にスプーンを持ってきているヤツがいたので、それでやってもらった。すると手でひねるように90度まで曲げ、あとはポトリと折ってしまった……超能力は存在する。誰にも文句はいわせない、ひえ〜。

前多 和洋 (16) 愛知県

う〜ん。キテます、キテます。

◆日経新聞 (4/30) によれば、通産省の諮問機関が5月に行う提言の中には透視力やテレパシーなどの超能力の解明があり、テレポーテーションの研究も候補に挙がっているとのこと。こ



人物の絵はまあ普通ですが、バックのこの感じがなかなかいいですね。これからどんどん送ってください。



岡村 直也 兵庫県
岡村くんはもう一通りコマングを送ってくれたんですが、両方ともなかなか面白かったです。雰囲気少し広告マンガっぽいですが。

これらのものの実現は2020〜2050年頃が目標ですが、いったいどんな社会になるのでしょうか。伊藤 進 (34) 愛知県

そりゃもう大変ですよ。なにしろ透視力でしょう。でへへ。

◆僕はマドルのイースIII冒険記がとっても大好きです。マドルのセリフはメチャ笑えます。そんな記事(?)を書く西川さんも好きです。ところで、今度Oh!Xに編集者さんたちの写真と簡単なプロフィールを載せてください。

細谷 和治 (14) 埼玉県

写真を載せてくれというハガキはよく来るんですが、そんなに见たいのならウチに就職すれば見られますよ。

◆GAME OF THE YEARプレゼントで「RPG秘宝館」が当たったんですが、「えっ? そんなもん応募したかな、ひょっとしてエッチなゲームソフトかしら……」とバックナンバーを確認してしまいました。しかし「秘宝館」って言葉にはあやしげな雰囲気がありますよね。伊豆地方にもいっぱいあるので一度見に来てください。

大野 二郎 (23) 静岡県

そうか、いっぱいあるんですか……。じゃあ秘宝館のハシゴでもしようかな。

◆ドラクエIVを解く前に最後のボスが読売新聞の日曜版に出ていたのを見てしまった。面白さが半減してしまった。くそー!

今井 勇樹 (17) 東京都

そうだったのか、見なくてよかった。私はまだ終わっていないのだ。

◆2月に受験のため東京に来てあの秋葉原に行きました。ある店でX1用アメリカントラック(新品)が50円で売られていたので思わず買いました。家に帰ってびっくり、テープ版だったのです(私はデータレコーダを持っていない)。ほかのソフトはディスク版ばかりだったじゃないか。

千葉 生樹 (18) 宮城県

50円じゃあしょうがないような気がしますけど。

◆池袋にある大手安売りショップに信長の野望を買いに行った。なぜかという、3月号の「CRISIS in Tokyo」と友人の家でやったボビュラスに影響されてシミュレーションがやりたく

なったからである。が、そこにX1シリーズのコーナーがなかったのである。店員に聞くとX1というパソコンさえ知らないというありさま。つくづく時代の流れを思い知らされました……うるうる。

北島 謙次 (17) 東京都

店員がX1を知らなかったというのがいちばん悲しい。まあ、たまたまそうだっただけのことだろうけど。

◆国際無線がまぶなくなって、そして千石通商が半分小さくなって、某量販店が建つことになってしまった。秋葉原ではあの通りがいちばん好きだったのに非常に残念だ。あれじゃ「再開発」の名を借りた「破壊」じゃないか。やっぱりアキバの魅力ってオールマイティじゃなくてアクの強さだと思う。

幅田 圭吾 (18) 茨城県

私はまだ秋葉原に行ったことないんです。大阪の日本橋なら何度も行ったことがありますけど。ああいうのは残してほしいですね。

◆祭りで買ってきた金魚はどうして長生きしないのか。また1匹……はかないものだ。

小林 宏昭 (17) 新潟県

僕が幼い頃にも買ってもらった金魚は鯉ぐらいの大きさになるまで生きたので、結局学校の池に放してあげました(ひょっとして金魚じゃなかったのかな)。

◆5月5日は僕の誕生日でした。10代最後の19歳となりました。来年はもうおじさんかあ。

井上 敬介 (19) 神奈川県

そうか、20歳以上っておじさんなのか。自分がおじさんになっていたなんて全然気がつかなかった。

◆「天国に一番近い」ニューカレドニア島の北東400kmのところにエロマンガ(Eromanga)島という島があるが、そこで日本語学校を開いたら島の住民は島の名前を変えるかもしれない。

松浦 範明 (16) 広島県

そういえば手塚治虫の「三つ目がとおる」で和登サンがエロマンガという女性と間違われる話があったなあ。

◆少し前のことだが「Japan Clasic」というアメリカの大学のバスケットボールのオールスター

戦をテレビで見ていたら、ベンチの控え選手がアルファエーを飲んでた。はるばるアメリカからやって来て試合中にあんなものを飲まされる選手たちは気の毒である。

松井 和宏 (21) 東京都

実は相手チームの陰謀だったらしい。

◆「PC-286対応S-OS「SWORD」(互換機可)」というところがやっぱりOh!Xのいいところ(?)だと思う。

三村 康夫 (19) 福岡県

いやあ、そんなにほめられると照れちゃうな。

◆結婚3年目を迎えようとするこの頃。嫁入り道具のひとつとしてやってきたX68000は、仕事の忙しい私の知らないところで妻の手により、すっかりゲーム専用機とされているようです。増えていくソフト、そして平成2年1月4日に長男誕生。X68000が母子により私の手から奪われる日は近い……? 宇崎 篤典 (28) 群馬県

もともとは奥さんのものなんですよ。いいじゃないですか。

◆私のX68000ACEは近くの工事のミスのため昇天してしまった。そして再起不能となってしまったので弁償してもらったことになった。ほかの家電のうち直せるものは直したうえでその費用を払ってもらったが、なにしろトランス(電柱についてるゴミバケツみたいなもの)が落ちて高圧電流が我が家をかけめぐったのだから愛機は再起不能となったのである。ところが買い換えようするとACEは流通在庫のみしかなく、それにEXPERT IIが出るのでそれ待ってはどうかとのこと、最近ようやく手に入れたのであった。

北風 裕介 (18) 兵庫県

ACEがEXPERT IIになったんだから、やっぱり得したということになるんだろうな。

◆X68000を買うまでずっと使っていたMZ-80Bを友達に譲ることにした。パソコンについての知識はまったくないそうだが、プログラムを作れるようになりたいといっていた。そこで、こっからこの話を持ちかけたわけである。MZにしてみればホコリをかぶっているよりも誰かに使ってもらったほうがしあわせですよなえ?

伊藤 盛人 (16) 福島県

そりゃそうですよ。ちなみに私もPC-6601

SRを友人にあげました。しかし、私自身は現在パソコンを持っていないのでした。

◆4月6日シャープからDMが届いた。いつもの渋い白い封筒ではなく派手な真っ赤な封筒だった。中を開けてみて「えっ、X68000に新機種!」とびっくりしました。まさか仕様は変えていないだろうな、たぶんそんなことはないと思うけど……。そして、おそろおそろ仕様表を見ました。「ゲツ、変えてる!」、その仕様表には「BIOSスピードアップ(平均2倍)」と書いてあったのです。シャープに文句言ってるやろうかと思いました。だってBIOSってROMでしょ、従来機のパンフレットにもそう書いてあるし。ROM変えたら交換のメンテナンスを受けるしかバージョンアップの方法はない。その日は落ち込んだ状態のまま寝ました。次の日、またシャープからDMが届いた。今度は例の渋い白い封筒だ。内容は春の見・体・験フェアのことだった。招待状も入っていた。よしこれに行って真相を明らかにしようと思った。4月14日学校の帰りにNSビルに寄りシャープの人にこのことを聞いてみた。答えは「ROMは一切書き換えていません。デバイスドライバの拡張です」とのことだった。僕にはその声が神様の声に聞こえた。

山口 隆久 (17) 東京都

このことについてはけっこうハガキが来ていたのですが、そういうことですからみなさん安心してください。まあ、5月号の記事の中に書いてあったのですが。

◆先日ショットバーでフィンランド製の缶ビールを見ました。そのラベルにはなんと東郷平八郎の写真が印刷されているのです。フィンランドと東郷平八郎の関係について誰か知っている方がいたら教えてください。

山崎 真二 (34) 埼玉県

というわけで知っている方がいたら教えてください。私は全然知りませんから。

◆1986年2月号よりずっとOh!MZ(Oh!X)を購読しています。連休中にじっくりと読み返してみましたが、参考となる記事の多さにあらためて「この雑誌は良い雑誌だ!」という認識を深めました。私のMZ-2500はメーカーのサポートこそなくなりましたが、3つのモードすべてでシ

ステムソフト(もちろんS-OS含む)が走るのでOh!X(THE SENTINEL)のあるかぎり成長を続けるたのもしいいマシンです。3モードそれぞれに個性があって面白いですよ。私もいつかは胸のすくような面白いソフトを投稿したいものです。編集部のみなさん、どうかこれからも魅力ある雑誌作り頑張ってください。それから健康にはくれぐれも気をつけてください。

春日 貴幸夫 (32) 鳥取県

どうもありがとうございます。そういっていただけるとこちらもすごくやる気が出ます。

◆そうですか、もう1000年になりますか。早いものですね。霧降高原からのYumiさんはお元気でしょうか。昔のOh!MZを見ていて、ふと目に止まるのはいつもこのページです。あれは1ページだけど全ページの中で唯一難しい話題から離れた「オアシス」のような存在でした。もしかしたらオアシスそのものだったのかもしれないね。いま読んでとても感じのいい文章で好きです。また良かったら再開してほしいですね(パロディも出たことだしね)。お願いします。

屋野 明正 (18) 大阪府

いまある記事の中でオアシスのような存在というところになるんだろうか。やっぱりそういうのは絶対必要ですよ。

◆スタッフになってみたいと思うのだからないやうにない。5年間パソコンを使っているけどもにプログラムも組めない(できない度95%)。また、東京近郊とも言え難い。茨城県でもかなり東京寄りなのだが最寄りの駅まで車で30分弱かかるのである。駄目押しは文章力がない。残念である。

倉田 泰幸 (20) 茨城県

本当に残念だなあ。スタッフになるといろいろいいこともあるのに、ふっふっふ。

◆5月号のちゃだワに載った睦地君、実は僕も最初の機械はM5だったのですよ。あの頃はよかった。ところであなたのM5どうなってます? 私のは完全に引退してベッドの下で眠ってますけど……。私は根っからのゲーマー、そしてホビーストなので間違ってもビジネスマシンであるPC-9801なんて買いません。私がパソコンに望むのは正にパーソナルワークステーション、つまり「自由に目的を決め、自由に使うことができるシステム」(取扱説明書より)。やっぱりいろいろやってみたいもんねえ、自分で。でもM5は惜しかった、俺大好きだったのにな。しかしX68000とM5って似ていると思いませんか? 「パソコン」のとらえ方が。

林 祐一 (19) 東京都

やっぱりみんな最初に使ったマシンというのは心に残るんですね。私もやっぱりPC-6001のことは忘れないでしょう。

◆いま思えばノーランズのセクシーミュージックが流行っていたあの頃(ちょっとずれてるかな)。MZ-80Kで「人間やめますか? それとも……」というほど面白かった。バグ・ファイヤーをやっていたあの頃「なににシャープ専門誌が出たんだって。女の人の絵が表紙があ、(裏



返して)ちょっと高いなあ買うのよそと)(ごめんなさい)という出会いから数えてついに通巻100号ですか。おめでとうございます。これからも楽しい記事を書かされるようがんばってください。

松澤 克明 (22) 東京都

さあ、いよいよ来月号で通巻100号です。記念になにかパーッとやりたいですね。

◆X68000のウイルス事件が新聞やテレビで取り上げられていましたが、いまいちわけのわからない報道ばかりでしたね。もう少し報道側もコンピュータについての予備知識くらい持っていてほしいですね。幸いサンデーネットなどにもワクチンがアップされていたので安心してところですが、ウイルスを作ること自体をなくせないものでしょうかね。カナシイ……。

佐藤 誠 (30) 東京都

まあ今度のことで某紙はかなりいろいろといわれているようなので、これからはもっと慎重な対応をするようになるでしょう。

◆X68000の現存数の過半数にウイルス汚染が発生したというニュースを見た。こういうときにこそユーザーあるいは次期ユーザーの方々が

団結し、このような問題をはねかえしX68000のポテンシャルと位置づけを高め、より多くの人々の支持を得る必要があると思う(逆境を乗り越えてこそ次のパワーを生むものです)。

秋保 三秋 (31) 埼玉県

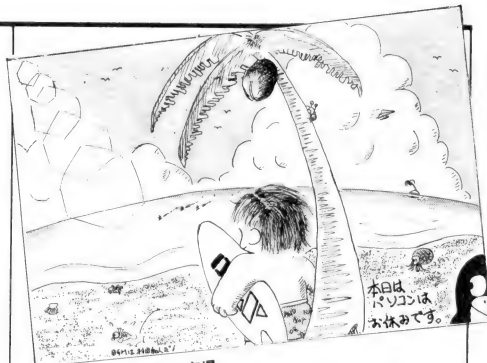
過半数がやられてるだって? どこのバカがそんなことをいったんでしょう。実際には1%にも満たないと思いますよ。

◆コンピュータウイルスのせいでX68000の人氣が落ちることはないでしょうか?

河野 幸孝 (16) 山梨県

ウイルスは大問題だけど、みんながしっかりとしていればそんなことにはならないと思います。どういうわけかかえって人氣が上がったりするんですよ。

◆1990年4月29日、私はなにげなく1989年5月号の「ちゃだワ」を読んでいた。そして、ふと目に止まった1通のお便り……。千葉県の石崎賢さん! 私小山田いくさんの大ファンなんです。それも「ぶるうピーター」の頃からの……。クラスの中にはファンの人がひとりもいなく淋しい私。どうか私と文通してください。



▲寺門 修司 兵庫県

なんかホノホノとしていいですね。やっぱりなんかホノホノとしていいですね。今年の夏はこういう誰もいない海岸へ行ってみたいな(無理かな)。

そして、いく先生の良さを語り合いたい。791-01 愛媛県松山市溝辺町47-1

住友 智代 (17)

この方は何通も同じ内容のアンケートハガキを送ってきてくれたので、しょうがないから最後にのせてあげました。どうかお手紙書いてあげてください。

ぼくらの掲示板

仲間

★このたびX68000ユーザーのサークルを作ることになりましたので会員を募集します。活動は月1回の会報の発行と情報交換などです。興味のある方は62円切手同封のうえご連絡を。初心者の方も大歓迎です。〒151 東京都渋谷区本町2-29-11コーポ丸山14号 高橋賢一郎 (18)

★当方ではX68000ユーザーを対象としたサークルを作るにあたってスタッフor会員を募集しています。内容はディスク会報の発行などをしていと思いますがそのほかにも会員の意見を取り入れたいと思います。初心者、女性大歓迎です。入会希望の方は62円切手同封のうえご連絡を。〒300-04 茨城県稲敷郡美浦村大谷246-1 小沢正人 (16)

★「NANNO CLUB PRO-68K」では、X68000ユーザーを対象に南野陽子のファンを募集します。ほかでは手に入らないSX-WINDOW用のアプリケーションを配布します。連絡は62円切手同封のうえ封書で。〒336 埼玉県浦和市常盤3-7-14 武山智裕 (18)

売ります

★CZ-855Dブラックを2万7千円、CZ-8BVIを5千円、CZ-8BSIを7千円、CZ-8RLIを3千円で。連絡はハガキで。〒526-01 滋賀県東浅井郡びわ町落合726 大森康雄 (23)

★マウスCZ-8NM2を3千円、プリンターケーブルCZ-800Pを3千円。エプソンVP-80K(トラクタ、PCROM付き)ヘッドピン折れのため漢字難ありを1万円(おまけX68000用ケーブル)。サイバーノート用電子手帳ADPを8千円で、すべて送料込み。連絡は往復ハガキで。〒281 千葉県千葉市三角町487-98 久吉一馬 (41)

★漢字プリンタ(XI turbo対応)エプソンRP-80F/TⅡKを送料別で1万5千円。新品同様、付属品一式、マニュアル付き。価格相談可。〒579 大阪府東大阪市昭和町1-7 青地信政 (36)

★(譲ります)「それ行け! XI」「それ行け! Xファミリー」誌、創刊3号から最終号まで20冊を無料で。ただし送料そちらもちで。いちばん欲しがっている方にさしあげます。必ず往復ハガキにてお願いします。〒104 東京都中央区晴海1-6-15-309 祐成好規 (30)

買います

★MZ-1500用RAMファイルMZ-IRI8、完動なら可。送料込みの希望価格を書いてハガキで連絡を。価格優先でおひとりに2週間以内に連絡いたします。〒581 大阪府八尾市弓削町1-191 小枝直隆 (19)

★XI用漢字ROMのCZ-8BK2またはCZ-8KKを6千円、XI用ディスクドライブCZ-503Fを1万2千円、XI用IOポートCZ-8EPを4千円で。いずれも送料込み。連絡は往復ハガキで。〒652 兵庫県

神戸市兵庫区下沢通2-3-4 吉岡延洋 (24)

★2400bpsのモデムを2万円で。連絡はメーカー・機種名明記のうえ往復ハガキで。送料はこちらもち。〒300-24 茨城県筑波郡谷和原村細代41-1 宮本健一 (19)

★XIで使用可能な24ピンドットインパクトプリンタを3万~5万円で。連絡は希望価格明記のうえ往復ハガキで。〒090 北海道北見市東陵町57-10高橋下宿内316号室 宮本康司 (19)

★XI用FM音源ボードCZ-8BSI+付属品を1万円以内で。連絡は往復ハガキで。〒999-35 山形県西村山郡河北町谷地字12堂16-15 斎藤一郎 (29)

バックナンバー

★Oh!MZ1986年3月号を送料込み2千円で買います。切り抜き不可。連絡は往復ハガキで。〒275 千葉県習志野市東習志野2-2-4-307 千田孝之 (20)

★Oh!XI1987年10月号、1988年3月号および9月号、1989年8月号を。読めれば傷・汚れは可。できれば4冊セットで。送料込み4千円で。連絡は往復ハガキにて。〒244 神奈川県横浜市戸塚区南舞岡2-12-3 野沢幸弘 (14)

★Oh!XI1988年12月号、1989年5月号を送料込みで各千五百円で。切り抜き不可。連絡は往復ハガキで。〒569 大阪府高槻市天神町2-30-8 有永聡 (21)

DRIVE ON

このコーナーでは、本誌年間モニタの方々のご意見を紹介しています。今回は、5月号の記事に関するレポートです。

●特集の「BASICプログラミング」についてですが、「みんながやってきたのだから、マニュアルを読め」という態度はよくないと思う。筆者はマニュアルを最初から1ページずつ読めと言っているのではなく、「使いたいコマンドがあれば、一度はマニュアルを引いてみなさい」という意味で言っているのだということもわかってはいます。しかし、あの文章を読んだだけでは「やっぱりこの味気ないマニュアルを読むのか……」と思います。自分が苦労してきたのであれば、あとから来る人には楽をさせてあげるのが務めというものだと思います。

湯澤聡 (27) X68000, XturboIII, MZ-286I/253I, PC-660I, MSX, PC-1360K 埼玉県

●特集ですが初心者には内容がちょっと……。やはり初心者に対してプログラミングを教えるためには、基本中の基本を教えるぐらいでないと。あと、ユーザー定義関数は一度使い方がわかるとすごく便利だと思う。友達のユーザーに教えてあげるがなかなかわかってくれない。自分は独学で使えるようになったが結構たいへんである。さらに、知っていれば便利なグローバル変数とローカル変数の違いも説明してほしい。

原田謙 (15) X68000PRO 広島県

●「潜入! バグ対策24時間、X-BASICはいま……!」は非常にわかりやすかった。個別の例をあげてくわしく説明してあり対策方法もわかりやすかった。プログラムを書くのはそれほど苦労しなくてもできるようになると思いますが、問題はそれあとのバグ取りです。これは実際に数多くこなさないとなかなか上達しない(正確なプログラムを書けばバグは

でないという話もあります)。こんなときのためにもバグ取りの記事などは読者にとってプラスではないでしょうか。特に最後のバグ取り最終奥義はおそらく実体験にもついたものでは? 自分の体験に重ねて非常に面白く読みました。

高田博 (32) X68000ACE-HD, PC-8001 和歌山県

●特集の中でいちばん面白かったのが「プロトタイピングのすすめ」です。いままでプログラムはフローチャートから組まなければスパゲッティになるといわれてました。しかし、この記事にはしろろとのプログラムはある程度大きくなってプロトタイピングがよいということが書かれていて賛成できました。

また、「ラジコンスティックの製作」ではパソコンとの接続うんぬんよりもラジコンの発信・受信の仕組みがわかったところがたいへん面白く読みました。今後はX68000ユーザーに限らず誰にでもできる回路の製作を期待したいです。

末吉克行 (21) XIG, MZ-73I, FM-7 兵庫県

●「ラジコンスティックの製作」はよい意味でマニアック過ぎる内容だったと思います。プロボを使ってアナログスティックを製作してしまうとは……。しかし、しかしです。私の住んでいる静岡というところは俗にいうナムコボタン(スイッチのこと)が1個350円もするのです。しかも、フットスイッチとなると……。これはもう6月号から始まる「ハードウェア工作入門」が低価格でできるものをやってくれることを祈るしかありません。

藤田康一 (19) X68000PRO 静岡県

●「言わせてくれなくちゃだワ」ですが、毎年ウケ狙いのハガキだけだと思っていたら結構マジメに考えている人もいるんですね(少数ですけど……。)そんな意見もたくさん載せてください。ところで、構成とかはよいのですがバックカラーをなんとかしてほしかった。136ページから137ページに目を移した瞬間、

チカチカときて……(私だけでしょうか)。

また、CARD.FNCを使ったカードゲームコンテストなんてよいのではないのでしょうか。

藤原博人 (26) XturboZ+NEW Z-BASIC 鳥取県

●「言わせてくれなくちゃだワ」はいつも楽しく読んでいます。Xユーザーの本音や思想などを知ることができるので、とても面白くと思います。また一緒に発表されている読者の所有機の変化も興味ある点です。いま現在、MZシリーズはどうなってしまうんでしょう……。半数がX68000のユーザーであるというのはとても凄いいことだと思ったりもします。今回は内容によって分類しているようですが、私は以前あったような地域割りのほうが親しみがあってよいように思いました。もっとも投稿者の住所が書いてあるのでこれでも十分なのはなのですが、その地域ごとの考え方の偏りのようなことは地域割りにしたほうが明確に表れて面白いように思います。今回は地域ごとの読者の割合なんかも出てませんでしたね。これは出ていてもあまり意味がないのかもしれませんが。

森川一 (24) X68000ACE-HD, XturboII 北海道

●前回のカードゲームのパターン定義をこのような形で発表してしまうとはまさに「必要は発明の母」といえるでしょう。CARD.FNCのおかげでX68000ユーザーはあまり苦労することなくカードゲームを数多く発表することができそうです。135Kバイトを5Kバイト近くまで(まあデータサイズをですが)、ごろうさまでした。たしかにカードゲームではパターンは同じものを使用してよいわけですし、それがきれいな絵柄であるならばとても喜ばしいことであります。スートのことなどうちくも面白くためになりました。BASIC特集にこれを早速活用したNinety-Nineが出ていたのもよかった。

大津和之 (20) XturboZ 福岡県

ごめんなさいのコーナー

6月号 付録ディスクの解凍

P.49 まとめて解凍する方法で、残ったSMPLを手動で展開してくださいというところの手順に間違いがありました。手順の最後の行は、
B>A: LH -E A: SMPL です。

また、HumanのシステムがVer.Iの場合は、おまけディスクをまとめて解凍することはできません。申し訳ありませんが個別に解凍してください。お詫びして訂正いたします。

バグに関するお問い合わせは
☎03(230)7683(直通)
月~金曜日 16:00~18:00

そのほかはP.130からのアフターケア参照。
6月号表紙絵の作者名が違っていました。作者は須藤牧人さんです。お詫びして訂正いたします。

お問い合わせは原則として、本誌のバグ情報のみに限らせていただきます。入力法、操作方法などはマニュアルをよくお読みください。また、よくアドベンチャーゲームの解答を求めるお電話をいただきますが、本誌ではいっさいお答えできません。ご了承ください。

Oh!X編集部は 7月16日より 新社屋に移転

▼今月の特集は久びさにマシン語特集となりました(1年半ぶり)。68000を中心としたマシン語入門ではZ80を中心としたマシン語入門とまた違った様相を呈していますね。今回はできるだけ初心者にも読みやすいようコンピュータ入門の立場からマシン語を紹介しようと、懐かしいぜんまいちゃんにも登場してもらいました。いかがでしたか。

また、8ビットのほうでは、ついにリロケータブルアセンブラWZDが完成、ZEDAやRED Aで飽きたらなかったS-OSユーザーの皆さんにも活用していただきたいと思います。

▼さて、いよいよ愛読者年間モニタを務めていただく方々のお名前を発表いたします。

浅野 憲(大阪府)、泉 昭彦(東京都)、奥村 光雄(埼玉県)、織田 聡(岐阜県)、高橋 毅(埼玉県)、高村 信(東京都)、谷口 正和(石川県)、段 宏太郎(兵庫県)、土谷 興正(兵

庫県)、中川 比呂志(東京都)、長谷川 敦士(山形県)、畑 剛志(北海道)、松井 伸康(東京都)、横山 賀一(東京都)、梅本 英之(奈良県)

以上15名(敬称略)の皆さんです。ささくですが今月号からモニタレポート用紙をお送りいたしますので1年間よろしく願いいたします。6月号までモニタを務めていただいた皆さんには心からお礼を申し上げます。

▼7月1日から株式会社日本ソフトバンクは「ソフトバンク株式会社」と社名を変更いたします。また、社屋も移転することとなり、これに伴いOh!X編集部の連絡先も7月16日から以下のとおりに変わります。

新住所 〒108 東京都港区高輪2-19-13
NS高輪ビル

☎03(5488)1309 (Oh!X編集部)

☎03(5488)1311 (バグに関する電話)

☎03(5488)1360 (出版営業部)

☎03(5488)1365 (広告営業部)

どうぞよろしくお願いいたします。

▼今月のマシン語プログラミング入門は筆者の村田敏幸氏急病のためお休みです。

投稿応募要領

- 原稿には、住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は、詳しい内容の説明、利用法、できればフローチャート、変数表、メモリマップ(マシン語の場合)に、参考文献を明記し、プログラムをセーブしたテープ(ディスケット)を添えてお送りください。また、掲載にあたっては、編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は、詳しい内容の説明のほか回路図、部品表、できれば実体配線図も添えてください。編集室で検討の上、製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして、他誌との二重投稿、他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒102 東京都千代田区九段南2-3-26井関ビル

日本ソフトバンク出版部

Oh!X「㊟㊟㊟」係

S H I F T ・ B R E A K

▶ジャッキー・チェンの番組を見ていた。BGMに聞き覚えがあると思ったら、三国志IIではないか。うーん、テレビ番組にはビックリだ。してみると、BGMとゲームミュージックは違うんだなあ。あ、向谷氏はFM音源のエキスパートでした。カシオペアファンの皆様、ごめんなさい。P.S.緑水Softwareの超高速の対応に感謝します。(H.U.)

▶神様、バンクしてしまいそうです。／もう一刻の猶予ありません。／神様、バンクしてしまいそうです。／サブ辞書が大きくなりすぎて、メモリからはみ出てしまいそうです。／この文章を打ちながら、何時くるかわからないエラーが怖いのです。／神様バンクしてしまうそうです。／はやく増設メモリを私の手に……。from君のソバで逢おう。(S.K.)

▶最近、統計分析をするんで某富士通の大型機を使っているのだが、パソコンとはずいぶん違った雰囲気なので苦労するのである。しかも、パソコンでできるようなことを、わざわざ大型でやるというのは非効率的だ。それもこれも、X68000じゃメジャーな表計算ソフトがあまりないからだ。ロータス1-2-3が欲しい……。 (龜)

▶ゲーム基板についての情報を兵庫県末光さんと神奈川県野沢さんほかからいただきました。どうもありがとう。先日私はナムコシステムIIを購入しました。しかし、かねてから腰を据えて遊びたかった「ワルキューレの伝説」がマニアたちの買い占めにあい、手に入りません。誰か売っているお店を知りませんか。(善)

▶なくなって初めてその存在の大きさがわかるものがある。水に空気にマウスのボタン。健在なときは誰もその大切さに気づかない。先日、バイクのチェンジレバーをふとしたはずみで折ってしまい、2日ばかり不便した。ギアチェンジができないのはつらい。たかだか大学→編集室間のなんと長く感じられたことよ(なら素直に電車使えよ)。(A.T.)

▶エラーがでた。エラーがでた。アセンブルエラーがでた。アセンブルエラーがでた。リンクエラーがでた。リンクエラーがでた。バスエラーがでた。バスエラーがでた。アドレスエラーがでた。アドレスエラーがでた。おかしい命令を実行しましたエラーもでた。おかしい命令を実行しましたエラーもでた。こんなにエラーがでるのは、カッコいい。(Mu)

▶ただし、とても恥ずかしいのだが、このさい連載も終わってしまったので書いてしまおう。実は私、マガジンに連載されていた「キラキラ」の愛読者だったのである。いい作品とはいえないが、作者(および主人公)の「屈折の仕方」に妙に共感を覚えるのだ。あーゆー屈折した漫画って少ないでしょ。若いバンドにはいくつかあるけど。(K)

▶またも薬師丸ひろ子の映画ははずれだった。しかし、斉藤由貴の映画は当たりが多い。「優駿」はともかく、最近の「君は僕を好きになる」と「香港パラダイス」はどちらも傑作だ。特に「香港パラダイス」は喜劇ながらラストシーンでは思わず涙した。涙もろいのは単に年をとったからではないと思う。

(Field of DreamsはつまらなかったKO)

▶編集部の雰囲気にもなじみ、やっとい息ついたり思っていたところ、バグ電話をとるようにといわれてしまった。前にも書いたように、ぼくはほとんど初心者だから困ってしまう。特に先月はディスクが付いたから山のように電話がかかってくる。初歩的なものは答えられるようになったけど、難しい内容のものはちょっと……。ごめんなさい。(A)

▶自分でいうのもなんだが、私は衣装持ちだ。「ロック少女風」「お嬢さま風」などその日の気分で着けている。そんな私をほかの部署の人間いわく「あれは“衣装”じゃなくて“仮装”っていうの」。ふうん、そっか、私は毎日仮装して会社に来ているのか。じゃ、せっかくだからゴジラのぬいぐるみでも……。ソフトバンクっていい会社だなあ。(E.O.)

▶パスワードを忘れる。盾の名前を片っ端から入れる。すぐに飽きる。Paint Mapを選ぶ。敵のパラメータをvery fastにする。敵の大陸を地震で均す。気候を砂漠に変えて敵の民を20人増やしてみる。……勝てない。増やすのを10人にする。……また負けた。しかし、人間の対戦を見てしまうとCONQUESTモードのなんと甘いことか。(U)

▶本誌の編集者は伝統的に、男子は西日本。女子は関東出身者で構成されている。というわけで、先月はウィルスのせいでろくに紹介できなかったA君だが、これがしゃべるしゃべる。以前かまた氏がいていた関西弁強化バックをつけているのではと思うほど。私なんかせっかく覚えたはずの東京言葉がわやくちゃんになっちゃった……。?? (T)

microOdyssey

コンピュータというものに出会ったのはいつのことだったろう。おそらく、中学生の頃に友人からコンピュータの雑誌を見せてもらったのが最初だと思う。当時、ぼくは何人かの仲間とデパートの屋上で「トランライザーガン」などのアーケードゲームに興じていた。だから、そのとき、「これを入力すればただでゲームができるのか」と思った。

数日後、ぼくたちは違うデパートにマイコン売り場があることに気づいた。遊び場はそこへと変わった。売り場といってもそのころはマイコンを買う人はあまりいなかったので展示場という感じだ。マイコンを一般の人にも知ってもらうということが主たる目的なのだ。

そこにはいろいろなマイコンがずらりと並んでいて、「どうぞご自由におさわりください」とでもいわんばかり。MZ-80K、ベーシックマスターなどがあり、ぼくたちはそこへ雑誌を持ち込んでプログラムを打ち込むようになった。

閉店時間までが勝負だった。あるのは本体のみでセーブはできないから。初心者ばかりだからいろいろなトラブルが起り、時間はどんどん消費される。エラーが発生するのは当たり前。マシン語のリストが出てくると「なんだ、この数字の羅列は。どうやって打ち込むんだ」。こういうことの連続で、なかなかプログラムを走らせるどころまでいかない。

しかし、ある日全部のプログラムを入れ終えることができた。そのプログラムはカマキリ星人が暴れ回るというようなゲームだったと思う。ぼくたちは焦った。すでに閉店直前だったのだ。まだ、マシン語部分のチェックサムの確認が済んでいなかったが、ぼくはたぶん間違えていないと思ったので「とりあえず走らせてみよう」と言った。なかには反対するものもいたが、やはり画面を見てみたいという欲求には勝てずRUNさせることになった。しかし、ぼくたちが見ることができたのは止まったままのカマキリ星人だけ……。ひとりがつぶやいた。「そういえばマシン語って暴走するんだよね」。

そんなこんなでぼくの趣味の項目にマイコンというものが加わった。高校の入学祝いにプログラム電卓「fx-602P」を買ってもらい、そのあとはPC-6001、PC-6601SRと所有機種を変え、Z80のマシン語も使えるようになった。でも、それからしばらくして情熱は薄れてしまった。いろいろな理由があったと思う。

マイコンと距離を置いたまま何年かの月日は流れたが、大学卒業後ソフトバンクに入社した。やはりコンピュータとのつながりは切れずに残っていたらしい。いや、未練があったというほうが正しいかもしれない。さらに、Oh!Xに配属されバックナンバーを読んでいると、こんなぼくを再びマイコンの世界に(趣味として)引きずり込むような記事を発見した。「S-OS」とMZ-700版「ゼビウス」である。

はつきりいって感動した。自分はなんてあきらめが早かったんだろうと反省もした。PC-6601SRは友人にあげてしまったのだが、もっともつつかいこめばよかった。そして、いまぼくはパソコンを買おうと思い、何にするか悩んでいる。マックにしようかな、Amigaにしようかな、と……。だって、一番好きなX68000はすでに会社の机の上にあるんだもん。(A)

1990年8月号7月18日(水)発売

特集 ADVANCED 2D GRAPHICS

Oh!X通巻100号記念特別企画

通巻100号記念特大プレゼント

特別企画 実況対戦ポピュラス 祝一平vs西川善司

全機種共通システム WLK

X68000用カードゲーム HEART

バックナンバー常備店

東京	神保町	三省堂神田本店5F 03(233)3312	神奈川	厚木	有隣堂厚木店 0462(23)4111
	//	書泉ブックマートB1 03(294)0011		平塚	文教堂四の宮店 0463(54)2880
	//	書泉グランデ5F 03(295)0011	千葉	柏	新星堂カルチェ5 0471(64)8551
	秋葉原	T-ZONE 7Fブックゾーン 03(257)2660		船橋	リプロ船橋店 0474(25)0111
	八重洲	八重洲ブックセンター3F 03(281)1811	//	//	芳林堂書店津田沼店 0474(78)3737
	新宿	紀伊国屋書店本店 03(354)0131	千葉		多田屋千葉セントラルプラザ店 0472(24)1333
	高田馬場	未来堂書店 03(200)9185	埼玉	川越	黒田書店 0492(25)3138
	渋谷	大盛堂書店 03(463)0511		川口	岩瀬書店 0482(52)2190
	池袋	リプロ池袋店 03(981)0111	茨城	水戸	川又書店駅前店 0292(31)0102
	//	西武百貨店9F コンピュータ・フォーラム 03(981)0111	大阪	北区	旭屋書店本店 06(313)1191
神奈川	横浜	有隣堂横浜駅西口店 045(311)6265		都島区	騒々堂京橋店 06(353)2413
	//	有隣堂ルミネ店 045(453)0811	京都	中京区	オーム社書店 075(221)0280
	藤沢	有隣堂藤沢店 0466(26)1411	愛知	名古屋	三省堂名古屋店 052(562)0077
北海道	室蘭			//	パソコンΣ上津津店 052(251)8334
			刈谷		三洋堂書店刈谷店 0566(24)1134
			長野	飯田	平安堂飯田店 0265(24)4545
					室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は、とじ込みの振替用紙の「申込書」欄に何年何月号からをご記入のうえ、年間購読料6,720円(税込)を添えてお申し込みください。その際、裏面の通信欄に「〇年〇月号よりOh!X定期購読希望」と忘れずに明記してください。なお、すでに定期購読をご利用いただいている方には、購

読期限終了と同時にご通知申し上げますので、同封の払込用紙をご利用ください。

海外送付ご希望の方へ

本誌の海外発送代理店、日本IPS(株)にお申し込みください。なお、購読料金は郵送方法、地域によって異なりますので、下記宛必ずお問い合わせください。

日本IPS株式会社

〒101 東京都千代田区飯田橋3-11-6

☎03(238)0700



7月号

■1990年7月1日発行 定価560円(本体544円)

■発行人 孫 正義

■編集人 橋本五郎

■発売元 (株)日本ソフトバンク

■出版事業部 〒102 東京都千代田区九段南2-3-26 井関ビル

Oh!X編集部 ☎03(230)7681

出版営業部 ☎03(230)7670 FAX 03(262)8397

広告センター ☎03(297)0181

■印刷 凸版印刷株式会社

©1990 SOFTBANK CORP. 雑誌 02179-7 本誌からの無断転載を禁じます。落丁・乱丁の場合はお取り替えいたします。

BEEP!

POWERFUL MEGA-MAGAZINE

MEGADRIVE

ビーブ!メガドライブ

7月号 定価480円(税込)
好評発売中

総額

200

万円相当

フランスウィック
ゲームアイデア
大募集!

特集

ゲームセンター改造計画

そろそろ「うるさい、雰囲気がよくない、
オタクのたまり場」のゲームセンターにさよならしよう!

ゲーム業界リクルート大作戦

もうすぐリクルートの夏がやって来る。
ゲーム業界志望の諸君、準備は大丈夫?

熱血メガドライブ宣言

ナムコ、金子製作所

徹底マスター

サンダーフォースIII フェリオス 大旋風
ムーンウォーカー バットマン



■特別付録
BEEP!メガドライブ
HALF YEARカレンダー

ゲームボーイ専門誌 パワーアップした第2弾だ!

ゲームボーイLIFE VOL.2

定価380円(税込)

54本のソフトを総ガイド

輝け! 第1回ゲームボーイ大賞

試験にでないゲームボーイ講座

業界初の完全攻略 オールソーサリアンシリーズ

FALCOM MAGAZINE [ファルコム・マガジン]

予価630円

オールアバウト・ソーサリアン パソコン版ソーサリアン
の総ガイドに加え、メガドライブ版ソーサリアンも紹介

オールファルコム・ベスト10 ファルコムユーザー100人
が選んだファルコムなんでもベスト10

日本ソフトバンク

日本ソフトバンクの 書籍特約書店

下記の書店の一覧は、日本ソフトバンク書籍特約店として右にある商品の他、新刊もとりそろえております。ご希望の商品がある場合は、下記のお近くの書店にてお買い求め下さい。

(注) 現品が売れて補充中の場合もございますので、ご注意ください。



日本ソフトバンク出版事業部

〒102 東京都千代田区九段南2-3-26 ☎03(230)7670

全国特約書店一覧



北海道	紀伊國屋書店札幌店	011-231-2131
札幌市	旭屋書店札幌店	011-241-3007
〃	丸善札幌支店	011-241-7252
〃	リーブルなにな	011-221-3800
〃	富貴堂札幌パルコ店	011-214-2303
〃	ダイヤ書房本店	011-712-2541
〃	ダイヤ書房西店	011-655-6223
旭川市	旭川富貴堂	0166-26-3481
〃	ブックス平和マルカツ店	0166-23-6211
〃	旭屋書店苫小牧店	0144-36-5185
苫小牧市		
青森市	成田本店	0177-23-2431
〃	岡田書店	0177-23-1381
弘前市	紀伊國屋書店弘前店	0172-36-4511
〃	ブックイン城東	0172-28-2882
八戸市	伊吉書院	0178-44-1917
盛岡市	東山堂書店本店	0196-53-6464
〃	さわや書店	0196-53-4411
〃	第一書店	0196-53-3355
仙台市	金港堂	022-225-6521
〃	金港堂ブックセンター	022-223-0979
〃	アイエ書店駅前店	022-264-0718
〃	九善仙台北支店	022-266-1127
〃	高山書店	022-263-1511
〃	ブックスみやぎ	022-267-4422
秋田市	三浦書店	0188-33-8131
山形市	八文字屋	0236-22-2150
福島市	岩瀬書店コルニエツタヤ店	0245-21-2101
〃	博向堂	0245-21-1161
郡山市	東北書店	0249-32-0379
いわき市	ヤマニ書房本店	0246-23-3481
〃	鹿島ブックセンター	0246-28-2222
会津若松市	宝文館	0242-27-5198
原町市	文芸堂	0244-22-1720
水戸市	川又書店駅前店	0292-31-0102
〃	ツルヤブックセンター	0292-25-2711
勝田市	武石書店	0292-73-1212
東海村	大野書店	0292-82-2098
鹿島郡	なみき書店	0299-96-1855
土浦市	共栄堂	0298-21-6134
つくば市	丸善筑波大学会館店	0298-51-6000
〃	友朋堂吾妻本店	0298-52-3665
宇都宮市	落合書店オリオン店	0286-34-3777
〃	落合書店東武ブックセンター	0286-34-8271
〃	新星堂宇都宮店	0286-33-2337
小山市	進路堂駅前ビル店	0285-25-1522
前橋市	煥平堂	0272-23-1211
〃	リプロ前橋店	0272-34-1011
〃	戸田書店前橋店	0272-61-5063
高崎市	学陽書房	0273-23-4055
〃	サカサ書店	0273-62-1500
〃	新星堂高崎店	0273-27-3961
〃	戸田書店高崎店	0273-63-5110
太田市	ナカムラヤ	0276-22-2001
浦和市	須原本店	048-822-5321

浦和市	須原店コソソ店	048-824-5321
大宮市	押田謙文堂	048-641-3141
〃	ブックセンター押田	048-647-3141
〃	三省堂ブックポート	048-646-2600
蕨市	須原原蔵店	0484-44-1211
川口市	岩瀬書店川口店	0482-52-2190
川越市	黒田書店川越店	0492-25-3138
所沢市	芳林堂所沢店	0429-25-5355
〃	いけだ書店所沢店	0429-28-3271
上福岡市	黒田書店上福岡店	0492-66-0120
朝霞市	文教堂朝霞店	0484-76-0107
志木市	新星堂志木店	0484-74-0182
春日部市	文教堂春日部店	048-752-7666
比企郡	錦電サービス	0492-96-2962
千葉市	多田屋セントラルプラザ店	0472-24-1333
〃	キディランド千葉店	0472-25-2011
習志野市	蔵翠堂	0474-72-5011
船橋市	ときわ書房本店	0474-24-0750
〃	リプロ船橋店	0474-25-0111
〃	旭屋書店船橋店	0474-24-7331
〃	芳林堂大井町店	0474-78-3737
〃	芳林堂津田沼店	0474-65-0926
〃	第二蔵翠堂	0474-65-0926
〃	三省堂書店西船橋店	0474-34-3111
柏市	西アサノ	0471-44-2111
〃	新星堂柏店	0471-64-8551
松戸市	堀江良文堂	0473-65-5121
〃	辰正堂駅ビル店	0473-64-7997
横浜市	有隣堂トーヨー店	045-311-6265
〃	有隣堂東ルミネ店	045-453-0811
〃	栄松堂相鉄ジョイナス店	045-321-6831
〃	せこうブックセンター	045-465-2111
〃	丸善ブックメッツポルタ店	045-453-6811
〃	有隣堂伊勢佐木店	045-261-1231
〃	有隣堂戸塚店	045-881-2661
〃	文華堂戸塚店	045-864-5151
〃	アーバン文華堂	045-821-5151
〃	文教堂青葉台南口店	045-983-5150
川崎市	有隣堂アゼリア店	044-245-1231
〃	有隣堂川崎BE店	044-200-6831
〃	文教堂本店	044-244-1251
〃	文教堂清ノ口店	044-811-8258
鎌倉市	島森書店大船店	0467-46-3841
〃	鎌倉書店	0467-46-2619
横須賀市	平坂書房WALK店	0468-25-5537
藤沢市	有隣堂藤沢店	0466-26-1411
〃	リプロ藤沢店	0466-27-0111
〃	文教堂六会店	0466-82-9610
茅ヶ崎市	川上書店ルミネ店	0467-87-3827
平塚市	サクラ書店駅ビル店	0463-23-2751
〃	文教堂四之宮店	0463-54-2880
小田原市	八小堂書店	0465-22-7111
〃	伊勢治書店	0465-22-1366
〃	文教堂小田原店	0465-36-3677
厚木市	有隣堂厚木店	0423-23-4111
大和市	文教堂中央林間店	0462-75-4165
相模原市	文教堂相模大野店	0427-49-0650
〃	文教堂橋本本店	0427-74-5581

相模原市	文教堂星ヶ丘店	0427-58-6121
津久井郡	文教堂城山店	0427-82-9278
東京		
千代田区	三省堂書店神田本店	03-233-3312
〃	書泉グランデ	03-295-0011
〃	東京堂書店	03-291-5181
〃	旭屋書店水道橋店	03-294-3781
〃	丸善お茶の水店	03-295-5581
〃	蔵翠堂	03-291-1362
〃	いずみ神田南口店	03-254-8521
〃	明正堂秋葉原店	03-257-0758
〃	T-ZONE	03-257-2660
中央区	八重洲ブックセンター	03-281-1811
〃	日本橋丸善	03-272-7211
〃	旭屋書店銀座店	03-573-4936
港区	書原新橋店	03-591-8738
〃	雄峰堂N S店	03-503-6586
〃	虎ノ門書房本店	03-502-3461
〃	虎ノ門書房田町店	03-454-2571
品川区	芳林堂大井町店	03-474-4946
〃	明屋書店五反田店	03-492-3881
渋谷区	紀伊國屋書店渋谷店	03-463-3241
〃	旭屋書店渋谷店	03-476-3971
〃	三省堂書店渋谷店	03-407-4545
〃	大盛堂書店	03-463-0511
〃	紀伊國屋書店笹塚店	03-485-0131
新宿区	紀伊國屋書店本店	03-354-0131
〃	三省堂書店新宿西口店	03-343-4871
〃	福家書店センタービル店	03-345-1246
〃	福家書店野村ビル店	03-342-0298
〃	新星堂N Sビル店	03-344-2055
〃	西武新宿ブックセンター	03-208-0380
〃	芳林堂高田馬場店	03-208-0241
〃	未来堂	03-200-9185
豊島区	旭屋書店池袋店	03-986-0311
〃	芳林堂池袋店	03-984-1101
〃	リプロ池袋店	03-981-0111
〃	三省堂書店池袋店	03-987-0511
〃	新栄堂本店	03-984-2345
〃	新栄堂アルパ店	03-988-0181
台東区	明正堂中通り店	03-831-0191
墨田区	ブックストア・談	03-635-1841
葛飾区	文教堂青戸店	03-838-5938
江戸川区	文教堂西葛西店	03-689-3621
大田区	アクトブックスサンカマタ店	03-735-1551
〃	竜文堂大森駅前ビル店	03-775-3851
中野区	明屋書店東京本社	03-387-8451
杉並区	ブックセンター荻窪	03-393-5571
〃	書原杉並店	03-313-4778
武蔵野市	紀伊國屋書店吉祥寺東急店	0422-21-5543
〃	弘栄堂吉祥寺店	0422-22-1031
〃	バルコブックセンター吉祥寺	0422-21-8122
調布市	真光書房	0424-87-2222
府中市	啓文堂	0423-66-3151
三鷹市	三省堂書店三鷹店	0422-48-4510
〃	東西書房	0422-46-0275
小金井市	文教堂小金井店	0423-86-0161
国分寺市	三成堂国分寺店	0423-25-3211

展示図書一覧

定価は本体価格です。

MS-DOSいたれりつくせり本 ●1800円
 プレイMS-DOS ●1900円
 UNIX System V
 プログラマ・ガイド ●12000円
 UNIX System V
 ユーザ・ガイド ●9800円
 UNIXオペレーティングガイド ●3000円
 OS/2 APIブックI ●2709円
 C言語の活用理解 ●2000円
 C言語の基礎知識 ●2500円
 C言語の応用50例 ●2300円
 上級・C言語の応用例50例 ●2400円
 Cプリプロセッサ・パワー ●2200円
 Play the C 上・下 ●各1500円
 Turbo C入門 ●2600円
 C++プログラミング ●2600円
 Quick Cプログラミング ●2602円
 詳説C言語 ●4369円
 8086アセンブリ言語 ●2800円
 8086マクロプログラミング ●2600円
 Final Ver.4.0ブック ●2400円

MIFES Ver.4.0ブック ●2400円
 ビジネスソフトデータ活用ブック ●2800円
 BASICによるプログラミング
 スタイルブック ●1800円
 ソーティング・ノート ●1900円
 J-3100パワーユーザーブック ●2400円
 続・PC工作入門 ●1800円
 PC-286Lブック ●1700円
 試験に出るX1 ●2800円
 RDBファラオ活用ガイド ●2903円
 言図ガイド ●2301円
 Rydeenガイド ●2427円
 P1EXEガイド ●2524円
 Lotus1-2-3ガイドII ●2500円
 MS-Chart Ver.3.1ガイド ●2900円
 まいと〜くガイド ●2300円
 新松ガイド ●2000円
 一太郎Ver.3ガイド ●2500円
 新一太郎ガイド ●2300円
 桐Ver.2ガイド ●2500円
 花子応用ガイド ●2500円

Lotus1-2-3ガイド ●2400円
 P1ガイド ●2300円
 NinJa2 ガイド ●2300円
 Multiplan
 Ver.3.1ガイド ●2400円
 アセンブラCASL入門 ●2000円
 ハードウェア徹底マスター ●2500円
 FORTRAN徹底マスター ●2800円
 情報処理の基礎知識 ●1600円
 COBOL 徹底マスター ●2900円
 受験用語ハンドブック ●1800円
 情報処理入門1・2 ●各1204円
 CASLで学ぶ
 アセンブラ言語入門 ●2204円
 バイト&ワードの風について ●1800円
 田原総一郎のパソコンウォーズ ●1400円
 パソコンを襲う
 知的独占の戦い ●1600円
 RPG幻想事典・日本編 ●1800円
 魔法王国シムルグント ●1800円

国 立 市 東西書店 0425-75-5061
 小 平 市 文教堂小平店 0423-43-9229
 東村山市 文教堂東村山店 0423-96-1115
 立 川 市 オリオン書房ウイ路店 0425-27-2311
 町 田 市 くまざわ書店本店 0426-25-1201
 有隣堂町田店 0427-23-3018
 久美堂本店 0427-25-1330
 久美堂小田急店 0427-27-1111
 文教堂鶴川店 0427-35-4117
 文教堂小川店 0427-96-1781
 多 摩 市 くまざわ書店桜ヶ丘店 0423-37-2531
 福 生 市 文教堂福生店 0425-53-7708
 <甲信越・北陸>
 甲 府 市 文教堂甲府店 0552-22-4600
 長 野 市 平安堂長野店 0262-26-4545
 長谷川書店 0262-26-2122
 上 田 市 平安堂上田店 0268-22-4545
 松 本 市 ブックスロクサン 0263-35-5555
 改造社松本駅前店 0263-36-3777
 飯 田 市 平安堂飯田店 0265-24-4545
 岡 谷 市 笠原書店 0266-23-5070
 諏 訪 郡 平安堂下諏訪店 0266-28-1111
 新 潟 市 紀伊屋書店新潟店 025-241-5281
 萬松堂 025-229-2221
 北光社 025-228-2321
 長 岡 市 覚張書店 0258-32-1139
 ブックセンター長岡 0258-36-1360
 長岡技大長峰文化 0258-46-6437
 バストピア コスモス 0255-25-5867
 BOOKメディア 0254-77-3850
 富 山 市 柳川書店 0764-24-4566
 清明堂 0764-24-4166
 BOOKSなかだ豊田店 0764-32-1353
 文苑堂本郷店 0764-22-0552
 文苑堂赤江店 0764-33-0321
 文苑堂 0766-21-0333
 文苑堂横田店 0766-21-0431
 うつのみや片町店 0762-21-6136
 書林香林坊本店 0762-20-5011
 野々市町 王様の本店 0762-46-5325
 福 井 市 勝木書店 0776-24-0428
 品川書店新田塚店 0776-24-1112
 <東 海>
 静 岡 市 静岡谷屋屋敷町本店 0542-54-1301
 江崎書店 0542-54-4481
 吉見書店 0542-52-0157
 戸田書店S.B.S店 0542-81-5733
 戸田書店曲金店 0542-81-5899
 沼 津 市 吉野屋 0559-23-5676
 マルサン書店宝塚店 0559-63-0350
 富 士 市 戸田書店富士店 0545-51-5121
 清 水 市 戸田書店本店 0543-65-2345
 浜 松 市 浜松谷屋屋敷本店 0534-35-9121
 名古屋 三省堂書店名古屋店 052-562-0077
 星野書店近鉄ビル店 052-581-4796
 丸善名古屋支店 052-261-2251
 丸善ブックメイテントラルパーク 052-971-1231
 日進堂上津店 052-263-0550

名古屋 三洋堂パソコンショップΣ 052-251-8334
 三洋堂いりなか本店 052-832-8202
 ちくさ正文館本店 052-741-1137
 白樺書房西店 052-774-7223
 豊 橋 市 精文館 0532-54-2345
 岡 崎 市 ブックス鎌倉 0564-54-1822
 豊 田 市 三洋堂梅坪店 0565-35-2334
 豊 川 市 三洋堂豊川店 05338-3-0334
 刈 谷 市 三洋堂刈谷店 0566-24-1134
 春日井市 三洋堂勝川店 0568-32-7806
 岐 阜 市 自由書房 0582-65-4301
 大 垣 市 大洞堂ブックス258 0584-81-2553
 大洞堂岐大バイパス店 0584-74-7766
 一 宮 市 三洋堂一宮店 0586-77-5734
 可 見 市 三洋堂可見店 0574-63-2334
 多治見市 三洋堂多治見店 0572-24-0340
 津 市 別所書店IIビル店 0592-24-1014
 四日市市 文化センター白揚 0593-51-0711
 鈴 鹿 市 シェア白揚スズカ 0593-82-3277
 <近 畿>
 京 都 市 駿々堂京宝店 075-223-1003
 アバンティ・ブックセンター 075-682-5031
 オーム社書店河原町店 075-221-0280
 ジュンク堂京都店 075-252-0101
 オーム社書店竹田店 075-644-2611
 奈 良 市 駿々堂大丸店 0742-26-6241
 大 阪 市 旭屋書店本店 06-313-1191
 紀伊屋書店梅田店 06-372-5821
 オーム社書店大阪店 06-345-0641
 駿々堂京橋店 06-353-3209
 駿々堂心斎橋店 06-251-0881
 旭屋書店ナンバ店 06-644-2551
 ナンパブックセンター 06-644-5501
 ヒバリヤ書店ナンバ店 06-644-5407
 旭屋書店アペノ店 06-631-6051
 ユーゴー書店 06-623-2341
 河村書店 06-951-2968
 枚 方 市 水嶋書房京阪デパート店 0720-51-3432
 高 槻 市 コーベックス西武高槻店 0726-83-1766
 大阪 市 ヒバリヤ書店本社 06-722-1121
 神 戸 市 ジュンク堂センター街店 078-392-1001
 ジュンク堂サンバル店 078-252-0777
 海文堂書店 078-331-6501
 日東館書林 078-391-8701
 姫 路 市 新興書房 0792-85-3344
 誠心堂書店 0792-81-2055
 和歌山市 宮井平安堂 0734-31-1331
 帯伊書店 0734-22-0441
 <中 国>
 岡 山 市 紀伊屋書店岡山店 0862-32-3411
 丸善岡山支店 0862-31-2261
 津 市 津山ブックセンター 08682-6-4047
 広 島 市 紀伊屋書店広島店 082-225-3232
 丸善広島支店 082-247-2251
 金正堂 082-248-3715
 積善館 082-248-3151
 尾 道 市 啓文社尾道店 0848-37-5151
 福 山 市 啓文社福山店 0849-22-3111

福 山 市 ブックシティ啓文社 0849-25-0050
 啓文社コア 0849-41-0909
 山 口 市 五十部誠文堂 0839-24-6630
 文栄堂 0839-22-5611
 下 関 市 中野書店 0832-22-6181
 宇 部 市 京屋書店 0836-31-2323
 本広書店 0836-31-0086
 防 府 市 誠文堂国術店 0835-25-1988
 光 市 三文字屋 0833-71-0251
 鳥 取 市 富士書店 0857-23-7271
 松 江 市 園山書店 0852-21-4167
 <四 国>
 徳 島 市 小山助学館本店 0886-54-2135
 小山助学館東口店 0886-25-1380
 森住丸善 0886-23-3228
 高 松 市 宮脇書店本店 0878-51-3733
 丸 亀 市 宮脇書店丸亀店 0877-22-5533
 松 山 市 紀伊屋書店松山店 0899-32-0005
 明屋書店本店 0899-41-4141
 明屋書店大街道店 0899-41-4242
 丸三書店 0899-31-8501
 新居浜市 明屋星原店 0897-44-4000
 宇和島市 明屋宇和島店 0895-23-1118
 高 知 市 金高堂 0888-22-0161
 <九州・沖縄>
 福 岡 市 紀伊屋書店福岡店 092-721-7755
 リーぶる天神 092-713-1001
 横文館新天町店 092-781-2991
 福岡金文堂本店 092-741-2106
 福岡金文堂朝日ビル店 092-431-1094
 福岡金文堂デイトス店 092-451-6175
 福岡金文堂アニマート原 092-844-0088
 ナガリ書店 093-521-1044
 北九州 金栄堂 093-531-3685
 旭屋書店北九州店 093-631-6421
 井筒屋ブックセンター 093-641-0131
 カルバーク平野 093-661-7988
 白石書店本城店 093-601-2200
 エマックスたがみ 0942-33-1841
 久留米市 BOOKリード 0948-25-7266
 大 分 市 バルコブックセンター大分店 0975-35-0663
 本町見星堂 0975-33-0231
 別 府 市 明林堂 0977-23-2183
 宮 崎 市 中央・田中書店 0985-24-3511
 寿屋宮崎店 0985-27-4111
 佐 賀 市 金華堂北バイパス店 0952-32-1965
 横文館佐賀店 0952-24-4314
 横文館デイトス店 0952-23-7155
 長 崎 市 メトロ書店 0958-21-5453
 好文堂 0958-23-7171
 佐世保市 金屋書店 0956-22-4241
 熊 本 市 紀伊屋書店熊本店 096-322-5531
 長崎書店 096-353-0555
 人 吉 市 明屋人吉店 0966-22-5486
 鹿児島市 春苑堂ブックプラザ 0992-25-3200
 ブックスみすみ 0992-57-1011
 那 覇 市 球陽堂書房ビル店 0988-63-3752
 文教図書 0988-62-1201

緊急発売!

究極の万能ワクチンに誕生!

サイバーワクチン いてこまし

対応機種: X68000 5インチ2HD

開発: 日コン連コンピュータウイルス研究所

¥3,000

【内容・特長】

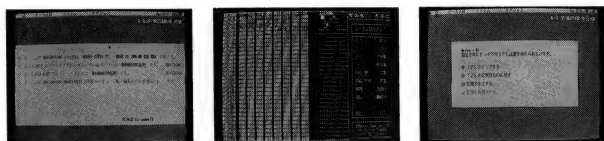
難波1号、難波2号など日コン連が確認している現存するすべてのウイルスの検知、除去が可能。特に、IPLから侵入してくるタイプ、システムプログラムに寄生するタイプ、難波1号・2号タイプのウイルスについては、さらに、ウイルスの検知、除去にいたるまでの完全自動化を実現。S-RAMに寄生するタイプについては、マニュアルによる検知、除去が可能。

フルマウスオペレーティングによる操作環境を実現。プルダウンメニューから選ぶだけで、初心者にもラクラク操作できる。

今後、新しいウイルスが発生した場合、個別形式で対応。

【仕様】

S-RAMの状態を表示する・ダンプする・クリアする・ファイルに落とす。システムのIPLをチェックする。システムのファイルをチェックする。メインメモリをダンプする。個別対応ワクチンを実行する。



■コンピュータウイルス学会発足

入会資格: 1959年以降に生まれた人。コンピュータウイルスについての解析、論述のできる人。詳細は、日コン連全国本部まで。

好評発売中! X68000用ソフト

X1ターボ版も誕生!宿題が楽になったと大好評!

翻訳ヘルパー ずるかまし ¥5,980

対応機種: X68000, X1ターボ, PC-9800, PC-8800, FM-TOWNS 各2枚組み

開発者: (X68000版) 大阪市立大学マイコン研究会 山本賢一

(X1ターボ版) STUDIO ATTIC 水無月みるく

辞書作成: 大阪市立大学マイコン研究会 山本博之

【内容・特長】

英文翻訳ガイド、英和辞書、和英辞書、英単語暗記トレーニング、辞書ユーティリティからなる翻訳の友。辞書4800語付。



発売15カ月経過、ただ今再びヒット上昇中!

D_RETURN

原作・開発者: 神戸大学情報統計部 赤坂賢洋

¥5,980

20カ月連続出荷達成!驚異のロングセラーソフト!

アドベンチャーゲームインタプリタ

電脳作家 Ver2.0

原作・開発者: 神戸大学情報統計部 村尾元

¥5,980

電脳作家グラフィック&ミュージックライブラリ集

制作者: 神戸大学情報統計部 細見格・赤坂賢洋

¥3,980

電脳作家シナリオ集①

EVIL EYE: 作 三上潤一郎

¥2,980

スターマンの伝説: 作 川合一広

日コン連SOFT保証

お客様のご都合により、同一種の新しいディスクとの交換を希望される場合には、そのディスクと360円分の切手をお送りください。折り返し、新しいディスクをお送りさせていただきます。

開発中! X68000用ソフト 各5,980円

近日発売!究極の3Dドライビングゲーム!

3Dで驚異のスピードを達成!

F. T. SCAN(エフ・ティ・スキャン)

開発者: Final Tear Z/Seafy-NAZ-Spark

お断り・・・X1ターボ版ずるかましの出荷本数が、X68000版ずるかましの10%に達しない場合、X1版のF. T. SCANの発売を取りやめさせていただきます。(5月末現在、X1ターボ版のずるかましの出荷本数は、X68000版の6%にしか達していません。)

太陽系を舞台としたわが国初のエコロジーシューティングゲーム

MEGA PRESSURE (メガ・プレッシャー)

開発者: 関西学院大学電脳研究会 池田尚隆

ビジュアルシーンふんだん、涙と感動のストーリー展開!

全150面 時間制限なしの究極の熱中型パズルゲーム

HOP UP (ホップ アップ)

開発者: 関西学院大学電脳研究会 池田尚隆・浅田真一・河野匡格

縦スクロールシューティングゲーム衝撃の超大作!

Task Force AL FARENE(タスクフォース・アルファ)

開発者: Shilpheed Soft 野村恵・磯野友厚・小村俊平

本格的ファンタジーアドベンチャーゲーム!

AQUARIUS(アクエリウス)

原作・開発者: 神戸大学情報統計部 赤坂賢洋

グラフィック: 神戸大学情報統計部 細見格・中野博之

お知らせ!

日本コンピュータクラブ連盟加盟団体募集中!

加盟費・会費不要。毎月、全国本部広報紙「つうてんかく通信」無料送付。

■日コン連では、以下のスタッフを求めています。

- ・日コン連全国本部(難波)、関東本部(自由が丘)付けスタッフ
- ・日コン連コンピュータウイルス研究所非常勤スタッフ
- ・パソコン雑誌「C・able」のライター及びエディター

【お問い合わせは、下記まで】

日コン連全国本部 06-644-6901(代)/日コン連関東本部 03-702-2891

■「サークル日コン連」(日コン連加盟)個人会員募集中!

入会金なし、年会費1,000円 特典多数

■コンピュータクラブ新設支援

大学内にコンピュータクラブを作りたいとお考えの方は、日コン連にご相談ください。クラブ運営のノウハウや機材の提供をさせていただきます。

日コン連加盟サークルのある大学・・・岩手医科、白鷗、東京水産、東京学芸、早稲田、法政、青山学院、昭和、成蹊、工学院、東京電機、横浜国立、名古屋工業、朝日、滋賀、京都、京都教育、立命館、京都産業、竜谷、大阪、大阪市立、関西、近畿、大阪電気通信、神戸、神戸商科、関西学院、神戸女学院、甲南女子、和歌山、和歌山高専、鳥取、島根、岡山、福山、高知、愛媛、九州工業、鹿児島

■パソコン&キャンパス雑誌「C・able」 近日創刊

ご希望の方は、切手360円分同封の上、お申し込みください。

郵送料貼付切手には、オール記念切手使用!

日コン連SOFT通信販売のご案内

現金書留、郵便振替(大阪5-4873 日コン連企画株式会社)、為替、定額小為替で、希望商品名、対応機種名、数量明記の上、お申し込みください。(送料はサービス。)

このうち、現金書留、定額小為替でお申し込みの場合には、例えば5,980円の商品の場合には、端数を切上げ6,000円分お送りいただいて結構です。この際のおつり20円は、商品発送時に同額の記念切手でお返しいたします。

●問い合わせ・申し込み先

日コン連企画株式会社または日本コンピュータクラブ連盟
〒556 大阪市浪速区難波中2-4-3 村上ビル
TEL 06(644)6901(代)

HOST

△ 68000 専用
多回線 ホストソフト

PRO-68K

ついに
登場!

3回線 / 9回線

きみも、今日から局長さん

HOST 9 PRO-68K 概要

対応回線数 1~9回線
使用モデム ATモデム MNP(RTS/CTS)可
通信速度 最大9600bps
会員数 *最大9999人
掲示板数 *最大40個
機能 電子掲示板・電子手帳・電子会議(チャット)・会員情報

これらは、コンフィグファイルで設定できます。

注1: *印について拡張を希望する場合は、プログラムの書き換えが必要になりますので、
別料金にて対応致します。当社までご相談ください。
2: 2回線以上で運用される場合は、CZ-6BF1(シャープ純正)が必要になります。
3: このホストはテキスト形式の転送方法を採用しております。

■特長

●各種設定のコンフィグファイル化。●RS-232C回線とは別にキーボードからのアクセス、ダウンロード、アップロードが可能。●モニタで、各チャネルのユーザーの打ち込んだコマンドや通信状態を確認。●各掲示板別にSIG、ボードパスの設定。●メンテナンス作業のオンライン実行。(ボードインテックス、メールインテックス)●オンラインサインアップ等、ゲストへの設定が可能。●通信サービスTTP対応。●行編集(オンライン簡易エディタ)機能。●その他、システムレベルで会員情報の変更が可能。タイムアウトによる回線切断。PDS専用掲示板の採用。(1書込中で、ドキュメントとテキストプログラムの分離)●接続MNPタイプの識別。●ログイン、ログアウト時間の記録。●非アクセス時のモニタ画面消去可能。

HOST 3 PRO-68K

機能は統べて、「HOST 9 PRO-68K」と同じですが、対応回線数が、1~3回線に制限されて、低価格でユーザーに供給します。

バージョンアップ (Ver1.10) サービス実施中

現在発売されています製品は、Ver1.10に変更になっています。お使いの製品がVer1.00のユーザーの方のために、バージョンアップサービスを実施しておりますので、お早目に、ユーザー登録葉書をお送り下さい。

Ver1.10へ無料交換を実施しております。

好評発売中

HOST 9 PRO-68K ¥59,800円

HOST 3 PRO-68K ¥39,800円

SPS-NET
TSUKUMO-NET モデル運用中!!

今、X68000の
通信が変わる!!!

たみのる

ユーザー重視の機能を搭載して

好評発売中
17,800円

24/31KHz
ディスプレイ
対応

2

「たみのる」が
装いも新たに
「たみのる2」として登場!
「たみのる」が
通信入門版なら
「たみのる2」は
マニアタイプの
通信ソフトです!!!

△ 68000 専用
パソコン通信ソフト

「たみのる2」はX68000用に製作された通信ソフトです。
X68000の機能を十分に引き出して、ユーザーのたがが簡単に
操作できるよう工夫・製作されています。



当社の製品は全国の有名デパート、パソコンショップでお求めになります。尚、お求めにならない場合は、郵便局にてお申し込みください。●口座番号 郡山5-12298
●加入者名(株)エス・ピー・エス ●金額 代金
に3%の消費税を加算した額 ●通信欄
(裏面)ご希望ゲームソフト名、数量、代金
合計、年齢、氏名、機種名、テープかディス
クの種類。(一週間以上かかりますので、
お急ぎの方は現金書留をご利用ください。
その場合、おつりのいらないようにお願
いします。

■表示価格に消費税は含まれておりません。

△ 68000
HOST PRO-68K 使用

SPS-NET TEL (0245)46-1167(代)

好評 / 一般回線
運営中 (9回線)
(4回線) / MNPクラス7

24時間運営 (N81 XN)
ゲストID (GUEST)

* GUESTアクセスは無料ですのでぜひ、
一度試してください。

入会方法 登録料 ¥3,000 (税別)
会費 無料

下記の用紙に直接記入するか又は、コピーして記入し、72円
切手同封の上、「SPS-NET係」までお送り下さい。届き次第、
仮登録を行いID発行後SPS-NET専用の郵便振込み用紙
ならびに運用の手引きをお送りいたします。それに従い、3ヶ
月以内に登録料 ¥3,000 (税別) を御入金下さい。
入金確認後正式会員として再登録します。

例◎ パスワード=SPS-NET

(8文字まで大小文字の識別あり)

◎本名=大和五郎(8文字まで)

◎ペンネーム=大ちゃん(4文字まで)

◎年齢=30(現在の年齢)

◎電話=0245-45-5777(市外局番から)

◎職業=株式会社エス・ピー・エス(16文字まで)

◎住所=福島市太平寺字町内5-3(24文字まで)

◎自己紹介=SPS-NETをよろしく

(24文字まで)

◎システム構成=X68000 ACE-HD MD2400B

(18文字まで)

信頼と実績のお店

BASIC HOUSE

今が
チャンス

X68000を御買上げの方にもれなく
下記X68000グッズのいずれか1つを
プレゼント!

- A. PROSTAFF ジャンパー
- B. X68000目覚し時計
- C. ツタンカーメンZIPPO
- D. ビジネスバッグ

PRO SHOP & STAFF
68000

サポート万全! 我々にお任せください!
ボーナス一括払・長期クレジットOKです!!

NEW **68000**
SUPER-HD



- 大容量80MB 3.5' HD内蔵
- SCSIインターフェイス標準装備
- 疑似マルチタスク マルチウィンドウを
実現した"SX WINDOW"を搭載
- 処理速度大幅向上(平均2倍)

特別価格にて予約受付中!

BASIC HOUSE 超特価
(限定品)



CZ-602C-GY+CZ-612D-GY
¥3□□,000
CZ-652C-GY+CZ-612D-GY
¥3□□,000

NEW **68000 EXPERT II**



CZ-613C
CZ-612D
CZ-8PC4
定価¥667,600
BASIC HOUSE特価

NEW **68000 PRO II**



CZ-663C
CZ-605D
CZ-8PC3
定価¥609,800
BASIC HOUSE特価

AMIGA

CALL



Macintosh

CALL



CZ-8PC3	BH特価
CZ-8PC4	¥99,800
CR-3415CL	¥109,800
CR-3410CL	¥98,000
VP-2050	BH特価
CZ-8NS1	¥188,000
GT-6000	¥178,000
GT-4000	¥198,000
HS-10R II	¥49,800
HXD040	¥118,000

HXD042	¥128,000
IT-X640	¥158,000
IT-X680	¥188,000
MD24FS5	BH特価
MD12FS	BH特価
XE-1 PRO	BH特価
CYBER STICK	¥23,800
CZ-6BF1A	¥38,000
CZ-6BG1	¥58,800
CZ-6BM1	¥26,800

SX-68M	¥19,800
C Compiler PRO-68k	¥39,800
Mu-1	¥19,800
マジックパレット	¥19,800
Zs STAFF PRO-68k	¥58,000
C-TRACE68	¥68,000
CARD PRO68k	¥29,800
CZ-6EB1	¥88,000
CZ-8NT1	¥13,800
AN-S100	¥36,600

全国どこでも発送可 長期クレジットOK 送料全国均一¥1,000 宅配便にて即日配送

株式会社計測技研

本社営業部/マイコンショップ/通販部
大田原営業所/マイコンショップ

宇都宮市竹林町503-1 TEL0286-22-9811 FAX0286-25-3970
大田原市美原1-13-4 TEL0287-23-5352 FAX0286-23-5364

マイコンショップ **BASIC HOUSE**

お申し込み・お問い合わせは **0286-22-9811(代)**

2枚のボードが1枚になった

KGB-X68PRK



絶賛発売中

広大なメモリ空間を実現する最大4Mバイトの
高速増設メモリ

高速演算を約束してくれる
**数値演算
プロセッサ**

- メモリアクセスノーズウェイトによる高速アクセス
- CZ-6BE2/CZ-6BE4/CZ-6BP1との混在が可能です
- 複数枚のKGB-X68PRKの実装が可能です
- ジャンパの変更により任意のアドレス空間にメモリの配置が可能です
- ジャンパの変更により数値演算プロセッサの1枚目2枚目/未使用の選択が可能です
- 1M/2M/3Mメモリモデルは購入後にメモリをボード上に追加可能です
- 数値演算プロセッサにはデバイスドライバ(FLOAT3X)が付属します

※写真はKGB-X68PRK-14です

※CZ-602C/CZ-612C以外の機種ではCZ-6BE1/CZ-6BE1Aを実装している必要があります
※メモリアクセスノーズウェイトのため拡張I/O BOXでは動作しません

製品価格一覧

KGB-X68PRK-01 ￥58,000 (1Mメモリ/数値演算プロセッサ無し)	KGB-X68PRK-11 ￥96,000 (1Mメモリ/数値演算プロセッサ付き)
KGB-X68PRK-02 ￥74,000 (2Mメモリ/数値演算プロセッサ無し)	KGB-X68PRK-12 ￥112,000 (2Mメモリ/数値演算プロセッサ付き)
KGB-X68PRK-03 ￥98,000 (3Mメモリ/数値演算プロセッサ無し)	KGB-X68PRK-13 ￥136,000 (3Mメモリ/数値演算プロセッサ付き)
KGB-X68PRK-04 ￥122,000 (4Mメモリ/数値演算プロセッサ無し)	KGB-X68PRK-14 ￥160,000 (4Mメモリ/数値演算プロセッサ付き)

購入後の増設費用

メモリ	
1Mバイト	￥24,000
2Mバイト	￥51,000
3Mバイト	￥76,000
数値演算プロセッサ	
MC68881RC16	￥38,000

充実のBASICHOUSEハードウェア&ソフトウェア

高速12BIT, 16CH A/Dコンバータボード(KGB-AD12) X1	￥118,000	高速12BIT, 4CH D/Aコンバータボード(KGB-DA4) X1	￥98,000
フォトアイソレーション16BITデジタル入出力ボード(KGB-PIO) X1	￥42,000	汎用ローコストA/D&PIOボード(KGB-X1S) X1	￥19,800
ハードディスクインターフェースボード(KGB-HDIF) X1	￥16,000	高速12BIT, 16CH A/Dコンバータ(KGB-X68ADC) X68000	￥128,000
アイソレーション16BITデジタル入出力ボード(KGB-X68PIO) X68000	￥68,000	64180CPUボードMach180(KGB-CPXB) X68000	￥98,000
ハンディプリンタ&インターフェース(HANDYPRINTjack) X68000	￥24,800	ローコストMIDIインターフェース(MELODY BOX) X68000	￥16,800
BASIC拡張関数パッケージ(B6-6301) ￥9,800	C言語ライブラリ(B6-6305) ￥6,800	BASIC拡張関数パッケージC言語ライブラリ付(B6-6306)	￥14,800
ディスクキャッシュ(B6-6304) ￥6,800	Toys & Tools (B6-6307) ￥6,800	アイコンエディタ(B6-6303) ￥4,800	CP/M68Kエミュレータ(B6-6302) ￥19,800

高解像度モードRGB出力をシャープ液晶ビジョンで投影!

Macintosh, X68000, PC-9801対応液晶ビジョンインターフェース
間もなく登場!

BASICHOUSE BBS TECOSYS NET

TEL 0286-27-1829 / 1200/2400ボー・MNPクラス5/8ビット/パリティ無し/X制御無し
ゲストIDなし(オンラインサインアップ)

全国どこでも発送可 長期クレジットOK 送料全国均一¥1,000 宅配便にて即日配送

株式会社計測技研

本社営業部/マイコンショップ/通販部
大田原営業所/マイコンショップ

宇都宮市竹林町503-1 TEL0286-22-9811 FAX0286-25-3970
大田原市美原1-13-4 TEL0287-23-5352 FAX0286-23-5364

マイコンショップ

BASICHOUSE

お申し込み・お問い合わせは ☎0286-22-9811(代)

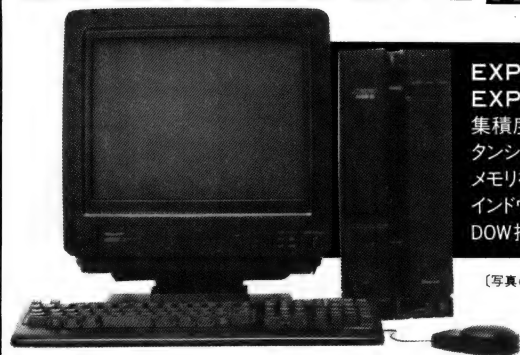


今すぐ もよりの電話から	仙 台 022-264-3704	名 古 屋 052-452-3271	広 島 082-295-6873
札 幌 011-611-5104	新 潟 0252-75-4175	大 阪 06-311-3931	福 岡 092-481-2494

X68000の情報のすべて!(当店はX68000の認定代理店です。お気軽にご相談下さい)

△ 68000 待望の新しい仲間登場!!

PERSONAL WORKSTATION
EXPERT II・EXPERT II HD



EXPERT II・EXPERT II HD
集積度を高めた"マンハッタンシェイプ"3Mの大容量メモリを搭載。本格的なウインドウシステム、SX-WINDOW搭載。

(写真のモニターは別売です。)

CZ-603C 標準価格 ¥338,000
CZ-613C 標準価格 ¥448,000

AVC 特価

△ 68000 待望の新しい仲間登場!!

PERSONAL WORKSTATION
PRO II・PRO II HD



PRO II・PRO II HD
拡張 I/Oポートを4スロットを搭載し、汎用性と低価格が魅力。もちろん、SX-WINDOW搭載。

CZ-653C 標準価格 ¥285,000
CZ-663C 標準価格 ¥395,000

AVC 特価

●但し消費税③%は別途請求させていただきます。●分割回数は3回〜48回まで自由に選べます。

X68000		お勧めディスプレイコーナー 組合せは自由、価格はお気軽にご相談下さい。			
 在庫稀少価格はお電話で!	CZ-604D 標準価格 ¥94,800 AVC 特価	●0.31mmドットピッチ ●2モードオートスキャン ●ステレオスピーカー搭載 ●チルト台同梱	CZ-613D 標準価格 ¥135,000 AVC 特価	●ドットピッチ 0.31mm ●TVチューナー搭載 ●ステレオスピーカー搭載 ●チルト台同梱	CZ-603D 標準価格 ¥84,800 AVC 特価
	CZ-602C CZ-612C CZ-652C CZ-662C	CU-21HD 標準価格 ¥148,000 AVC 特価	CZ-605D 標準価格 ¥115,000 AVC 特価	●ドットピッチ 0.39mm ●TVチューナー搭載 ●ステレオスピーカー搭載 ●チルト台同梱	CZ-602D 標準価格 ¥99,800 AVC 特価
		●0.52mmドットピッチ ●21型ディスプレイ ●3モードオートスキャン ●ステレオスピーカー搭載			●0.31mmドットピッチ ●TVチューナー無し ●3モードオートスキャン ●チルト台同梱
					●ドットピッチ 0.39mm ●TVチューナー搭載 ●チルト台同梱

型 番	品 名	標準価格	販売価格	型 番	品 名	標準価格	販売価格	型 番	品 名	標準価格	販売価格
CZ-6TU	システムチューナー	¥ 33,100	AVCフタバ特価	CZ-8PG1	24ピンカラープリンター (80桁)	¥ 130,000	AVCフタバ特価	CZ-8TM2	モデムユニット	¥ 49,800	AVCフタバ特価
BF-68PRO	CRTフィルター	¥ 19,800	AVCフタバ特価	CZ-8PK10	24ピンプリンター (136桁)	¥ 97,800	AVCフタバ特価	CZ-252MS	Musicstudio	¥ 28,800	AVCフタバ特価
CZ-8NS1	カラーキャプチャー	¥ 188,000	AVCフタバ特価	IO-735X	カラージェットプリンター	¥ 248,000	AVCフタバ特価	CZ-247MS	NEW Print Shop	¥ 19,800	AVCフタバ特価
CZ-6BN1	スキャナー用パラレルボード	¥ 29,800	AVCフタバ特価	CZ-6BE1A	1M増設 RAMボード	¥ 38,000	AVCフタバ特価	CZ-221HS	TOP給与計算機キースタート	¥ 200,000	AVCフタバ特価
CZ-6VT1	カラーイメージユニット	¥ 69,800	AVCフタバ特価	CZ-6BE2	2M増設 RAMボード	¥ 79,800	AVCフタバ特価	CZ-227BS	TOP財務会計	¥ 200,000	AVCフタバ特価
CZ-8BV2	カラーイメージボード	¥ 39,800	AVCフタバ特価	CZ-6BE4	4M増設 RAMボード	¥ 138,000	AVCフタバ特価	CZ-220BS	DATA	¥ 58,000	AVCフタバ特価
CZ-8BR1	立体映像セット	¥ 29,800	AVCフタバ特価	CZ-6BP1	数値演算プロセッサ	¥ 79,800	AVCフタバ特価	CZ-212BS	BUSINESS	¥ 68,000	AVCフタバ特価
CZ-8DT2	パーソナルテロップ	¥ 44,800	AVCフタバ特価	CZ-6BC1	FAXボード	¥ 79,800	AVCフタバ特価	CZ-219SS	OS-9	¥ 29,800	AVCフタバ特価
CZ-8BS1	FM音源ボード	¥ 23,800	AVCフタバ特価	CZ-6BM1	MIDIボード	¥ 26,800	AVCフタバ特価	CZ-211LS	Copier	¥ 39,800	AVCフタバ特価
CZ-8NJ1	ジョystick	¥ 1,700	AVCフタバ特価	CZ-6BL1	I/Oボード	¥ 39,800	AVCフタバ特価	CZ-234LS	AI-68K	¥ 188,000	AVCフタバ特価
CZ-8NM2A	マウス	¥ 6,800	AVCフタバ特価	CZ-6BL1	LANボード	¥ 268,000	AVCフタバ特価	CZ-620H	20MBハードディスク	¥ 178,000	AVCフタバ特価
CZ-8NM3	マウス・トラックボール	¥ 9,800	AVCフタバ特価	CZ-243BS	サイバーノート	¥ 19,800	AVCフタバ特価	CZ-64H	40MBハードディスク	¥ 120,000	AVCフタバ特価
CZ-6SD1	システムトラック	¥ 44,800	AVCフタバ特価	CZ-240BS	ステディショナリー	¥ 14,800	AVCフタバ特価	LHD-34V	40MBハードディスク (ロジック)	¥ 153,000	¥ 117,000
AN-S100	アンブレ内蔵スピーカー	¥ 36,600	AVCフタバ特価	CZ-223CS	通信ソフト	¥ 19,800	AVCフタバ特価	LHD-32V	20MBハードディスク (ロジック)	¥ 128,000	¥ 98,000
CZ-6EB1	拡張 I/Oボックス	¥ 88,000	AVCフタバ特価		ゲームソフト	20% OFF					

CZ-8NJ2



アナログジョystick
標準価格 ¥23,800

AVC 特価 ¥ ???

CZ-8PG2



24ピンカラー。
漢字ドットインパクト
プリンター

CZ-8PG2..... ¥160,000

AVC 特価 ¥ ???

CZ-8PC4

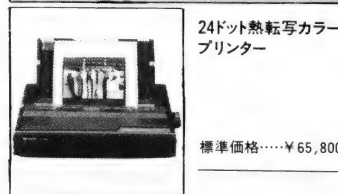


48ドット熱転写プリンター。
精密な文字、ハード
コピーも可能。

CZ-8PC4..... ¥ 99,800

AVC 特価 ¥64,800

CZ-8PC3



24ドット熱転写カラー
プリンター

標準価格..... ¥65,800

AVC 特価 ¥39,800

●頭金なし(手軽な電話クレジット) ●製品先取り(お支払いは約1〜2ヶ月後から) ●低金利クレジット(1回の支払いは2,700円以上で3〜48回。ボーナス併用可) ●カレッジクレジット(保証人なし。但し満20歳以上の学生の方) ●18歳未満の方(ご両親が代理購入者としてお申し込み下さい) ●納期(通常の場合、当社に申込書が到着後1週間以内。特に人気のある商品で品薄の場合、少々納期が遅れることがありますので御了承下さい) ●完全保証(すべてメーカー保証書付。アフターケア万全) ●全国代引(お届けした者に、代金をお支払いいただく方法です。但し手数料1,000円)

**AM10時からPM7時
まで受付 日曜・祝日も営業**

パソコン専門 O.A.ランド

アフターサービス万全
のサポート体制
優良パソコン販売店

- お近くの方は、お立寄り下さい。
専門係員がアドバイスいたします。
- ビジネスソフト、ゲームソフトのこと
ならおまかせ下さい!!

セール期間
◀ '90 6.15 ▶ 7.15

もうすぐ夏だヨ!! ドカ〜ンとプレゼント
OAランド恒例・大お買徳セール実施中

流通事情により、広告表示価格より、
お安くなる場合がありますので、ドンドンお電話下さい。

- 毎週日曜、第2・第4土曜日は、定休日と
させていただきます。

SHARP X68000シリーズセット (お楽しみゲームパック付)

X68000 EXPERT II

- CZ-603C-BK/GY
 - CZ-605D-BK/GY
 - MD-2HD 20枚
- 定価合計 ¥453,000



X68000 EXPERT II-HD

- CZ-613C-BK/GY
 - CZ-605D-BK/GY
 - MD-2HD 20枚
- 定価合計 ¥563,000

OAランド大特価

クレジット 12回 ¥30,200 24回 ¥15,900

NEW

OAランド大特価

クレジット 12回 ¥37,400 24回 ¥19,700

X68000 PRO II

- CZ-653C-BK/GY
 - CZ-605D-BK/GY
 - MD-2HD 20枚
- 定価合計 ¥400,000



X68000 PRO II-HD

- CZ-663C-BK/GY
 - CZ-605D-BK/GY
 - MD-2HD 20枚
- 定価合計 ¥510,000

OAランド大特価

クレジット 12回 ¥26,600 24回 ¥14,000

OAランド大特価

クレジット 12回 ¥33,800 24回 ¥17,800

X68000 SUPER-HD

- SX-WINDOW搭載
- SCSIインターフェース装備
- 80MBハードディスク搭載
- 3MB大容量メモリ装備
- 高解像度グラフィック



X68000 SUPER-HD

- CZ-623C-TN(チタン)
 - CZ-613D-TN(チタン)
 - MD-2HD 20枚
- 定価合計 ¥633,000

クレジット 12回 ¥40,600 24回 ¥21,400

NEW

OAランド大特価

X-1ターボZⅢセット

①セット

- CZ-888CBK...定価¥169,800
 - CZ-880DBK...定価¥109,800
 - CZ-6ST1B...定価¥5,800
(チルトスタンド)
 - MD-2HD 20枚サービス
- 合計定価¥275,400

②セット

- CZ-888CBK...定価¥169,800
 - CZ-830DBK...定価¥98,000
 - CZ-6ST1B...定価¥5,800
(チルトスタンド)
 - MD-2HD 20枚サービス
- 合計価格¥273,600



特価中TEL下さい

特価中TEL下さい

今月の特価品(限定)お早目に!!

★CZ-652C(BK)+CZ-602D(BK)

4セット限り...大特価¥258,000

- SHARP WD-A300(ワープロ) 定価¥165,000 特価¥115,000
- SHARP WD-A330(ワープロ) 定価¥185,000 特価¥129,000
- SHARP WD-HL30(ワープロ) 定価¥198,000 特価¥138,000
- SHARP PW-910(ワープロ) 定価¥85,000 特価¥85,000

★CZ-612C(BK)

3セット限り...大特価¥298,000

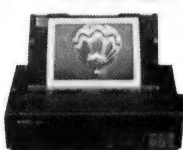
- NEC PC-KD853(アナログCRT) 特価¥50,000
- 三菱XC-1498C(アナログCRT) 特価¥54,800
- SHARP CU-14FD(アナログCRT) 特価¥46,000
- SHARP PA-8500(電子手帳) 特価¥16,800

周辺機器コーナー

プリンターセットコーナー

- CZ-6PVI(カラービデオプリンター) 定価¥198,000 特価¥152,000
- CZ-8PG3(24ドット熱転写カラープリンター) 定価¥65,800 特価¥53,000
- CZ-8PK10(24ピン漢字ドットプリンター・136桁) 定価¥97,800 特価/TEL下さい!
- CZ-8PGI(24ピンカラー漢字ドットプリンター・80桁) 定価¥130,000 特価/TEL下さい!
- CZ-8PG2(24ピンカラー漢字ドットプリンター・136桁) 定価¥160,000 特価/TEL下さい!
- IO-735X(カラーイメージジェットプリンター) 定価¥248,000 特価/TEL下さい!

OAランド特選品!!



■CZ-8PC4(定価¥99,800)

●48ドット熱転写カラー漢字プリンター 特価¥64,800

X68000用ソフトウェア・コーナー

- ①CZ-212BS(BUSINESS)...定価¥68,000 特価¥53,000
- ②CZ-220BS(DATA)...定価¥58,000 特価¥45,000
- ③CZ-215MS(Sampling)...定価¥17,800 特価¥13,800
- ④CZ-221HS(NEW Print Shop)...定価¥10,800 特価¥15,500
- ⑤CZ-227BS(TOP財務会計)...定価¥200,000 特価¥158,000
- ⑥CZ-226BS(GARD)...定価¥229,800 特価¥23,000
- ⑦CZ-223CS(Communication)...定価¥19,800 特価¥115,500
- ⑧CZ-213MS(MUSIC)...定価¥18,800 特価¥14,800
- ⑨CZ-211LS(C compiler)...定価¥39,800 特価¥31,000
- ⑩C-TRACE(キャスト)...定価¥68,000 特価¥52,000
- ⑪EW(イースト)...定価¥38,000 特価¥29,000

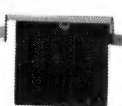
X68000用周辺機器コーナー

- CZ-6PU1A...定価¥38,000 特価¥30,000
- CZ-6BM1...定価¥26,800 特価¥21,000
- CZ-6BE1...定価¥88,000 特価¥69,800
- CZ-6VT1...定価¥69,800 TEL下さい!
- CZ-6NS1...定価¥188,000 特価¥149,000
- CZ-6BC1...定価¥79,800 特価¥63,000

- 最新ゲームソフト
その他各種ソフト
20%~25%OFF!!
- 周辺機器・プリンター
割引販売中!! TEL下さい!

■I・O DATA 増設RAMボード NEW

- 1MB増設RAMボード PIO-6BE1-A 定価¥25,000
- 2MB増設RAMボード PIO-6BE2-2M 定価¥50,000
- 4MB増設RAMボード PIO-6BE4-4M 定価¥88,000



特価¥19,500 特価¥38,500 特価¥67,000

■ハードディスク ■特価品もありますのでTEL下さい。

- アイテック ITX-640...特価¥117,000
- アイテック ITX-680...特価¥149,000
- ロジテック LHD-32V...特価¥85,000
- ロジテック LHD-34VE...特価¥90,000
- ロジテック LHD-34V...特価¥104,000
- シャープ CZ-620H...特価¥118,000
- シャープ CZ-64H...特価¥95,000
- アイテム HXD-040...特価¥88,000
- アイテム HXD-042...特価¥95,000
- ICM SR-80...特価¥130,000

中古パソコン

(価格/在庫は変動します。予約は5日以内とします。)

- PC-9801RA5...¥338,000
- PC-9801RA2...¥265,000
- PC-9801RX2...¥199,000
- PC-9801EX2...¥190,000
- PC-9801VX21...¥170,000
- PC-9801VX2...¥165,000
- PC-9801VX2...¥160,000
- PC-9801VM21...¥150,000
- PC-9801UV11...¥148,000
- PC-9801LV22...¥160,000
- PC-286VE...¥150,000
- PC-286US...¥155,000
- PC-286VS...¥165,000
- PC-600C...¥160,000
- CZ-601C...¥170,000
- CZ-611C...¥198,000
- CZ-652C...¥178,000
- CZ-612C...¥210,000
- 68000用モニター...¥49,000
- PC-9801用サウンドボード...¥13,000
- PC-88SR,FR...¥50,000
- PC-88FH,FA...¥65,000
- 400ラインCRT...¥38,000
- 200ラインCRT...¥10,000

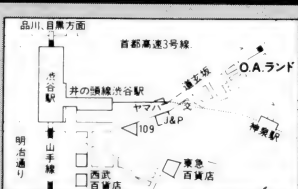
通信販売のご案内

全国通販

- 銀行振込で申し込みの方は商品名
及びお客様の住所・氏名・電話番号
をお知らせ下さい。

(振込先)第一勧業銀行 渋谷支店
普通No.1163457 株オーエーランド

- 現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1~60回払で月々5,000円より自由に設定できます。



- 取・買取は電話で見積りしております。責任を持って下取りさせて頂きます。
- ご注文、お問合せは... 午前10時から午後7時まで
- 商品のお届けは...入金確認後、即日発送致します。

株オーエーランド

〒150 東京都渋谷区円山町20-4 第5日新ビル1F

☎(03)770-8855 FAX (03)770-7080

関東エリアの送料は、1個につき¥1,000です。

- ★全商品保証書付。専門のアドバイザーが、お客様のニーズに対応します。
- ★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。

■表示価格は、税別表示です。詳しくは、お電話にて、お問い合わせ下さい。掲載の価格は、5月末現在です。

株式会社

デンキヤ



営業時間AM11:00~PM7:00 水・木曜定休

セット超特価

68000

PERSONAL WORKSTATION

PRO II・PRO II HD

CZ-653C特価

CZ-663C特価

SUPER HD

CZ-623C特価

CZ-613D特価

(価格はすべて税込みです)

セット超特価

68000

PERSONAL WORKSTATION

EXPERT II・EXPERT II HD

CZ-603C特価

CZ-613C特価

EXPERT PRO

CZ-662C特価

CZ-602C特価

全品メーカー保証 即決クレジットOK

ディスプレイ

プリンタ

周辺機器

ソフト

CZ-604D	特価	CZ-8PC4	特価	CZ-8NJ1	¥1,400	CZ-213MS	¥15,500
CZ-605D	特価	CZ-8PG1	特価	CZ-8NJ2	¥18,540	CZ-223CS	¥15,300
CZ-613D	特価	CZ-8PG2	特価	PIO-6BE1A	¥20,000	CZ-219SS	¥23,100
CU-21HD	特価	IO-735X	特価	PIO-6BE2	¥39,000	CZ-211LS	¥30,800

24時間テレホンサービス

0482-54-3444

お申し込み

TEL.0482-54-3400

FAX.0482-54-3443

埼玉県川口市西川口4-6-4

お支払い

下記取引銀行口座
までお振込み下さい。
三菱銀行西川口支店
(株)デンキヤ 0258081

X68000 新型発売記念 特価セール

△68000 EXPERT/PRO



CZ-602C (本体)
+ CZ-603D (ディスプレイ)
+ SX.WINDOW
大特価 ¥310,000
(このセットに限り、送料+消費税込)
CZ-653C (本体)
+ CZ-602D (ディスプレイ)
大特価 ¥288,000
(このセットに限り、送料+消費税込)

※代金は商品引換兼払いでもOKです。

■New X68000新発売/ (●特価価格は直接お問合せください。)

CZ-603C	定価 ¥338,000	<ディスプレイ>	●CZ-602.603
CZ-613C	定価 ¥448,000	CZ-603D	単体でも大
CZ-623C	定価 ¥498,000	CZ-604D	特価/ お問
CZ-653C	定価 ¥285,000	CZ-605D	合せてください。
CZ-663C	定価 ¥395,000	CZ-613D	

SHARPフリーストップパソコン

All in Note

AX286N-H2



定価 ¥398,000 → 特価

ALBIT

アイビット電子株式会社

FM TOWNS お買い得セット

1. TOWNS-1 (本体) ¥338,000
2. FMT-ME (1M) (増設メモリ) ¥ 60,000
3. FMD-FD301 (増設FDユニット) ¥ 28,000
4. FMT-KB101 (キーボード) ¥ 20,000
5. FMT-DP531 (カラーディスプレイ) ¥ 89,800
6. TOWNS-OS V1.1 L20 ¥ 20,000

定価合計 ¥555,800

大特価/ ¥285,000

MZ2500下取り/ MZ2500からMZ2861 (定価

¥328,000)に買い替え下取後 特価 ¥165,000

CZ600C下取り/ CZ600CからCZ623 (X68000

SUPER)に買い替え下取後 特価 ¥300,000

ハガキもOK. New MZプリンタ

漢字カラー シャープMZ-1P22

漢字30字/秒高速印刷 ●MZ1P

17フルコンパチ ●5KBのバッファ

メモリ付) 対応パソコン: MZ2000,

2500, 5500, 6500シリーズ, X1シリ

ーズ, X68000シリーズ他

標準価格 ¥59,800 → 特価 ¥25,000

パソコンファクス MZ-1V01

“プリンタ・コピー・ファクス”

1台3役のスクレモノ

限定セット販売!

●MZ2500セット (インターフェースソフト付)

標準価格合計 ¥342,800 → ¥120,000

●MZ-1V01 (本体のみ)

標準価格合計 ¥278,000 → ¥98,000

シャープMZ-1X30 モデムホン

(X19上位機種)

●300/1200bps全2重通信対応

モデム内蔵 ●音声入出力端子

付 ●ダイヤルパルス/プッシュボ

タン対応 ●プッシュボタン音解析

機能 ●シャープ手帳, CCITT, V.22

bis通信手順サポート

標準価格 ¥98,000 → 大特価

パソコンと専用ワープロをひとつにした16ビット

シャープMZ-2861

ワープロソフト「書院28」

MS-DOS V3.1 装備

エミュレーションソフト搭載

定価 ¥328,000 →

大特価 ¥198,000

(ディスプレイ別)

MZ-2861用ソフト (UPシリーズ)

●IP-1251 (デスクトップ) 定価 ¥88,000 → 特価 ¥20,000

●IP-1253 (クッキー) 定価 ¥77,000 → 特価 ¥20,000

●IP-1254 (プランナー) 定価 ¥88,000 → 特価 ¥20,000

シャープMZ-2520

定価 ¥159,800 → 大特価 ¥78,000

《限定在庫限り》

'89プログラム大賞グランプリ受賞作

“HEAVY METAL” 搭載

PC-E500PJ

定価 ¥28,800

→ 大特価

●ご購入の方に「ボクンジャー特別号」を呈呈。

PC-500C各機/パソコンをつなぐインターフェースケーブル

CE-140T ¥8,800

拡張機器他

●シャープCZ-68M1 (256MB) ¥26,800 → ¥23,000

●シャープCZ-8GR (X1 GRAM) ¥32,000 → ¥12,000

●シャープCZ-8EB3 (I/Oボックス) ¥33,800 → ¥28,000

●シャープCZ-8BK3 (X1) ¥13,800 → ¥11,700

●シャープCZ-8BK4 (X1) ¥6,800 → ¥5,700

●シャープCZ-8BK2 (X1) ¥14,800 → ¥4,000

●シャープCZ-64H (ハードディスク) (CZ-602C専用) 特価

●シャープCZ-8NJ2 (256MB) ¥23,800 → ¥18,500

●シャープCZ-8BK4 (X1) ¥38,000 → ¥25,000

●シャープCZ-8IT (ハードディスク) ¥8,500 → ¥1,000

●シャープMZ-1U08 (1500, 2000) ¥25,000 → ¥12,000

●シャープMZ-1U03 (1500, 2000) ¥35,000 → ¥15,000

●シャープMZ-1X2Z モデムユニット ¥21,800 → ¥13,000

●シャープMZ-1R12 RAM ¥35,000 → ¥8,000

●シャープMZ-1E29 (MZ) ¥17,800 → ¥9,800

●シャープMZ-1E30 (MZ) ¥25,000 → ¥22,500

●シャープMZ-1U09 (2500) ¥9,000 → ¥7,200

●シャープMZ-1M03 (5500) ¥69,000 → ¥35,000

●シャープMZ-286C04 (2000) ¥18,000 → ¥8,000

●シャープMZ-8B104 (2000) ¥45,000 → ¥18,000

●シャープMZ-1R11 (5500) ¥80,000 → ¥30,000

●シャープMZ-1R24 (1500) ¥22,000 → ¥6,000

●シャープMZ-1R26A (2500) ¥13,000 → ¥12,800

●シャープMZ-1R27A (2500) ¥13,000 → ¥10,000

●シャープMZ-1R28A (2500) ¥13,000 → ¥10,000

●シャープMZ-1R29A (2500) ¥32,000 → ¥10,000

●シャープMZ-1T02 (2200) ¥19,800 → ¥8,500

●シャープMZ-1T03 (1500) ¥12,000 → ¥8,500

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープMZ-1X29 (2500) ¥13,800 → ¥11,000

●シャープCZ-503F ¥49,800 → ¥30,000

●シャープCZ-53F ¥19,800 → ¥9,800

●シャープCZ-300F (CZ-3PCM付) ¥13,000

ハードディスク

●アイテックIT-X640 ¥158,000 → ¥128,000

●アイテックIT-X68 ¥198,000 → ¥158,000

ディスクレイ

●富士通FMTV-153 ¥108,000 → ¥76,000

●シャープMZ-17 (5500) ¥124,000 → ¥63,000

●シャープMZ-1D-27 ¥120,000 → ¥79,800

ソフト

(X68000用)

●CZ-230AS ニュージーランド ¥8,800 → ¥7,040

●CZ-2310AS FULL THRTIE ¥8,800 → ¥7,040

●CZ-233AS PACMANIA ¥7,800 → ¥6,250

●CZ-222AS ARKANOID ¥7,800 → ¥6,250

●POPULOUS ¥9,800 → ¥7,850

●CZ-239AS THUNDARBLADE ¥9,500 → ¥8,000

(MZ-2500用)

●IP-1215 COBOL ¥13,800 → ¥11,700

●DANGER BOX ¥5,800 → ¥2,000

●EXTRA HYPER DISK MONITOR ¥10,000 → ¥8,500

●EXTRA HYPER DISK MONITOR ¥14,000 → ¥12,000

●FILE UTILITY (UT-25F) ¥6,800 → ¥6,000

●FREE CALL ¥6,800 → ¥1,000

●G-EDIT2500 ¥8,000 → ¥7,000

●H.S. コントローラー ¥9,600 → ¥8,500

●HuCAL 日本語 ¥45,000 → ¥15,000

●レイトスコープ ¥9,800 → ¥3,000

●レイトスコープ2 ¥5,800 → ¥1,000

●ザ・ブラックオニキス ¥7,800 → ¥3,000

●スーパー修理屋さん ¥12,000 → ¥10,200

●ムーンチャイルド ¥7,800 → ¥3,000

●英雄伝説サーガ ¥9,800 → ¥2,000

●五目並べ ¥4,800 → ¥2,000

●探検隊第2弾 ¥7,800 → ¥2,000

●プリントSHOP ¥9,800 → ¥8,500

●プリントSHOP ライブラリー ¥4,500 → ¥3,800

●プリントSHOP ライブラリー2 ¥4,500 → ¥3,800

(X1用)

●日本語ワープロ将軍X1 ¥34,800 → ¥29,000

●日本語ワープロ将軍X1 ¥19,800 → ¥16,800

●CZ-8WB5 X1 データベース ¥9,800 → ¥3,500

●3CP/M X1 3rd CPM ¥16,800 → ¥5,000

●CZ-8BK3 X1 第2水準ROM ¥13,800 → ¥11,700

●CZ-128SF X1 CP/M ¥13,800 → ¥11,700

●CZ-130SF X1 CP/M ¥14,800 → ¥12,500

●CZ-116LF X1 C ¥13,800 → ¥11,700

●CZ-117S X1 LOGO ¥18,800 → ¥13,200

●CZ-118LF X1 COBOL ¥13,800 → ¥11,700

●CZ-126LF X1 APL ¥13,800 → ¥11,700

(MZ-5500, 6500SOFT)

●MZ-22013 (MZ-5500MSDOS)

●MZ-22014 (MZ-5500TODAY)

●MZ-22023 (MZ-5500GW BASIC)

●MZ-22028 (MZ-5500GW BASIC)

●MZ-22025 (MZ-5500ワープロ)

●MZ-22029 (MZ-5500TODAY)

本体 ●シャープCZ-820, 822, 880, 881, MZ-3500,

2520, 2861, 2200, X68000, C-612, 662, 602, 652 ●

富士通FM 77AV.1, 77AV.2, 77AV.20, 77AV.40 ● NEC

PC-9801N ●東芝J310055

《全商品新品完全保証付》 ■シャープボケコン全商品販売中、カタログ 特価表ご請求ください(〒72)。

0426-45-3001~3

FAX.0426-44-6002

●営業時間/10:00

アイ・ツ- EXE CLUB

新規ユーザー・EXE会員大集合

- ★ X68000ユーザーニーズに対応したハード・ソフト・ウェア・周辺機器は全て展示しています。
- ★ 新製品情報・ユーザー同士の情報交換ができる、メンバー様の憩いのスペースです。
- ★ 大特価セール期間中X68000・ディスプレイ・プリンター御購入の方は全国どこでも送料無料!!
- ★ 遠くでなかなかお越し頂けない方にも通販専用TELで専門スタッフ(X68 PRO STAFF)が親切丁寧にお答えします。
- ★ X68000お買い上げの方、アイ・ツ-よりBigプレゼント。

X68000 オリジナルステッカー
X68000 フロッピータイトルシール
X68000 オリジナルテレホンカード
X68000 バッグ

お好きなもの2点
もれなくついてくる!!

★ 現在シャープX68000 EXE会員の方、おトモダチをご紹介下さい。ご購入成立時点でアイ・ツ-とシャープよりステキなプレゼント進呈中!!

★ アイ・ツ-メンバーズ優待制度実施

アイ・ツ-でX68000・及びソフトウェア周辺機器をお買い上げ頂きましたユーザー様にはオリジナルメンバーズカードを送付致します。メンバーズの方には楽しいパソコンライフをおくれますように最善のフォローをアイ・ツ-より提供します。

春は気分も新たにいろんなことをやってみよう。
そんなあなたにとっておきの贈り物をシャープから。

★ X68000新製品

Super-HD (CZ-623C)、EXPERT II (CZ-603C)
EXPERT II-HD (CZ-613C)、PRO II (CZ-653C)
PRO II-HD (CZ-663C) ソクソク入荷!!

X68000のことなら、なんでもご相談下さい。

旧型製品も格安にて
"御提供中"

NEW

X68000 誕生

PERSONAL WORKSTATION

SUPER・EXPERT・PRO

- SX-WINDOW 搭載 オリジナルウインドウシステム
 - 大容量メディア対応 SCSI対応、大容量H.D.D
 - AV機能強化 ビデオボード、音多・AVディスプレイテレビ
- 頂点を極める3大機能

SB会販売コンテスト・パソコン部門最高峰賞 シャープ販売第一位受賞感謝セール! 期間 6月18日~7月17日

FAX特集

SHARP FO-50 定価¥ 99,800
SHARP UX-10 定価¥ 128,000
SHARP UX-20 定価¥ 168,000

大特価にて
展示中!

只今お買上の方、絶対必要なロール紙プレゼント中!!
全国どこでも送料無料。カタログ希望の方、ハガキに御使用機種(CZ-...)とか住所、ご氏名、TEL、生年月日をお書きのうえアイ・ツ-EXE CLUBへご送付下さい!

X68000プロショップ(専門店)



パソコンミュージック
MIDI
X68000

音遊サウンドライブ

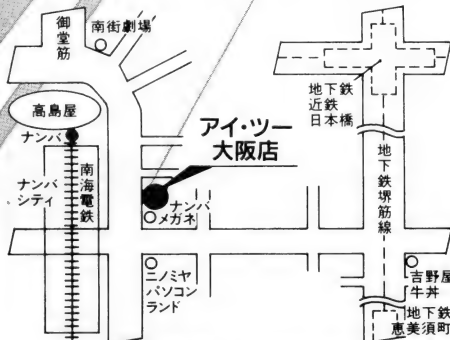
店頭イベント

X68000"MIDI"実体験

(パソコンができなくても、楽器ができなくても"MIDI"ができる)

店頭では"MIDI"だよX68000を切り口にパソコンミュージック"MIDI"が体験できる「X68000トレンド実体験コーナー」を設置X68000新製品(SUPER HD・EXPERT II・PRO II)ラインアップ展示でX68000の魅力を全てを訴求します。

逃がす手はない
チャンスです



■営業時間 AM11:00~PM8:00



アイ・ツ-inシャープグランドフェア'90 OSAKAスタジアムに多数のご来場頂きまして、誠にありがとうございました。アイ・ツ-サックスフェアPart2も只今企画中です。このご期待!!

X68000ユーザーとおきのグッズ!!

X68000ユーザーのステータスシンボル。

新グッズもグループインしてますます充実。

キミのパソコンライフが一層楽しくなるコレクションだ!

X68000オリジナルグッズをまだ持っていないキミ

アイ・ツ-からお届けしちゃうマス!



通販専用TEL.

06-633-9800

年中無休



Information & Interface
株式会社
アイ・ツ-

大阪店/〒542 大阪市中央区難波千日前15-18



クリエイイト特典

- 全商品完全保証書付(メーカー保証)
- 全国無料配達(一部離島の方は有料になります)
- 配達日の指定OK(日曜・祭日にかかわらずお客様のご都合にあわせて配達します)
- どんな商品の組合せも自由自在(ご予算、用途に応じ自由自在にシステムアップできます)
- 中古パソコン高額下取り(今お使いのパソコンをわずかな差額でグレードアップ)
- お支払い方法自由(低金利の均等払い、ボーナス一括払いもご利用ください)

営業時間(定休日▶渋谷店:日曜・祭日/横浜店:水曜)
AM10:00~PM7:00

当社はX68000の販売認定店です。
どんなことでも安心してご相談ください。

X68000
ボーナスセール開催中!!

即売・即納

X68000 NEW PRO II

- CZ-653C(本体).....¥285,000
- CZ-603D(カラーディスプレイ).....¥84,800
- 好きなゲームソフト1本.....¥7,800
- 定価合計.....¥377,600

クリエイイト特価

均等払い	¥7,680×48回	¥9,890×36回	¥14,370×24回
ボーナス	なし	なし	なし

X68000 NEW EXPERT II

- CZ-603C(本体).....¥338,000
- CZ-613D(カラーディスプレイテレビ).....¥99,800
- CZ-8NJ2.....¥23,800
- 好きなゲームソフト1本.....¥9,800
- 定価合計.....¥506,600

クリエイイト特価

均等払い	¥9,970×48回	¥12,840×36回	¥18,660×24回
ボーナス	なし	なし	なし

X68000 EXPERT II HD

- CZ-613C(本体).....¥448,000
- CZ-604D(カラーディスプレイ).....¥94,800
- 好きなゲームソフト1本.....¥9,800
- 定価合計.....¥552,600

クリエイイト特価

均等払い	¥5,920×48回	¥7,400×36回	¥12,100×24回
ボーナス	¥30,000×8回	¥40,000×6回	¥50,000×4回

X68000 SUPER HD

- CZ-623C-TN(本体・キーボード・マウス).....¥498,000
- CZ-613D-TN(カラーディスプレイ).....¥135,000
- CZ-6BP1.....¥79,800
- 定価合計.....¥712,800

クリエイイト特価

均等払い	¥7,320×48回	¥10,100×36回	¥13,450×24回
ボーナス	¥42,000×8回	¥50,000×6回	¥80,000×4回

※本広告に掲載の全商品の価格について消費税は含まれておりません。

X68000 NEW EXPERT II

ミュージシャンセット。これもTMネットワークだよ〜!

- CZ-603C.....¥338,000
- CZ-605D.....¥115,000
- MU1.B(MIDIボード&ソフト).....¥39,800
- CM32L.....¥69,000
- グラナダ.....¥8,800
- JOYカード.....¥1,800
- 定価合計.....¥572,400▶超特価¥458,000

X68000 NEW PRO II

ゲーマーズセット。遊んで暮らせるSET!

- PRO II CZ653C.....¥285,000
- 0.31CRT CZ603D.....¥84,800
- グラナダ.....¥8,800
- Y'S.....¥8,700
- ボビュラス.....¥9,800
- スーパーハンガオン.....¥8,800
- エージャックス.....¥8,800
- サーク.....¥8,800
- アールタイプ.....¥7,800
- アナログJOYSTIC XE-1AP.....¥13,800
- 定価合計.....¥445,100▶超特価¥353,000

X68000シリーズ用 周辺機器・ソフト オール超特価!!

型番	品名	定価	ソフト名	品名	定価
CZ-6VT1	カラーイメージユニット	¥69,800	MUSIC PRO	MIDI版	¥28,800
CZ-8NS1	カラーイメージスキャナ	¥188,000	MUSIC PRO-68K	マウスを使った楽譜ワープロ	¥18,800
CZ-6BE1A	1MB増設RAMボード	¥38,000	SOUND PRO-68K	サウンドエディタ	¥15,800
CZ-6BE2	2MB増設RAMボード	¥79,800	Sampling PRO-68K	AD PCMサンプリングエディタ	¥17,800
CZ-6BE4	4MB増設RAMボード	¥138,000	Musicstudio PRO-68K V.1.1	MIDIマルチレコーディングソフト	¥28,800
CZ-8NM3	マウス・トラックボール	¥9,800	OS-9/X68000	マルチタスクオペレーティングシステム	¥29,800
BF-68PRO	高性能CRTフィルター	¥19,800	PRO-68K	サイバーノート	¥19,800
CZ-6BP1	数値演算プロセッサ・ボード	¥79,800	PRO-68K	ステーションリ	¥14,800
CZ-8NT1	トラックボール	¥13,800	Ccompiler PRO-68K	ソフト開発セット	¥39,800
CZ-6BM1	MIDIボード	¥26,800	Human 68K Ver2.0	開発ツールセット	¥9,800
CZ-8NJ2	アナログスティック	¥23,800	PIO-6BE1-A	内蔵1MRAM	¥25,000
CZ-6TU	パソコンチューナ	¥33,100	PIO-6BE2-2M	2MRAM	¥50,000
SX-68M	MIDI I/F	¥19,800	PIO-6BE4-4M	4MRAM	¥88,000
XE-1AP	アナログジョイパッド	¥13,800	MU1-B	MIDI I/F + ソフト	¥39,800

▲上記以外ビジネスソフト、最新ゲームソフト豊富に在庫あります。※送料はご注文の際にお問合せください。●超特価販売中!

オール15%~20%OFF

総合お問合せ先 ☎03-486-6541代

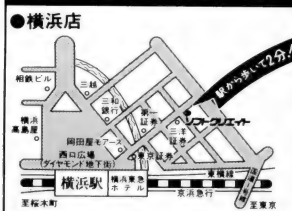
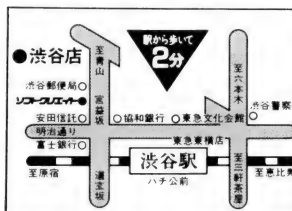
パソコン専門ショップ

ソフトクリエイイト 渋谷/横浜

●渋谷店 ☎03-486-6541(代) 〒150:東京都渋谷区渋谷1-12-7 三和渋谷ビル
振込銀行:三井銀行 渋谷宮益坂支店(No.5000340)

●横浜店 ☎045-314-4777(代) 〒221:横浜市神奈川区鶴屋町2-12-8 第1建設ビル
振込銀行:三和銀行 横浜駅前支店(No.310852)

★この表以外の組合せ、お支払い方法もご自由にできます。
★X1シリーズ用、X68000シリーズ用各社ハードディスク/プリンタ等の周辺機器を大特価にて販売しております。
電話にてお問合せください。





パソコンミュージック **MIDI**
68000

音遊サウンドライブ

こんどの6月24日(日)

MIDIって何だ? というあなたも、コンピュータミュージックならまかせろ/という君も、思わずナットク。スタジオでの多重録音による音楽作成の現場をプロミュージシャンがトークショー形式ですべてお見せします。
こんなチャンスはざらにない!



ADO・TOYOMURA **T-ZONE** ティー・ゾーン

Micom Zone

2F 〒101 東京都千代田区外神田4-4-1 ☎257-2650

海外でも使える

「T-ZONE CLUB」

カード会員募集中!!

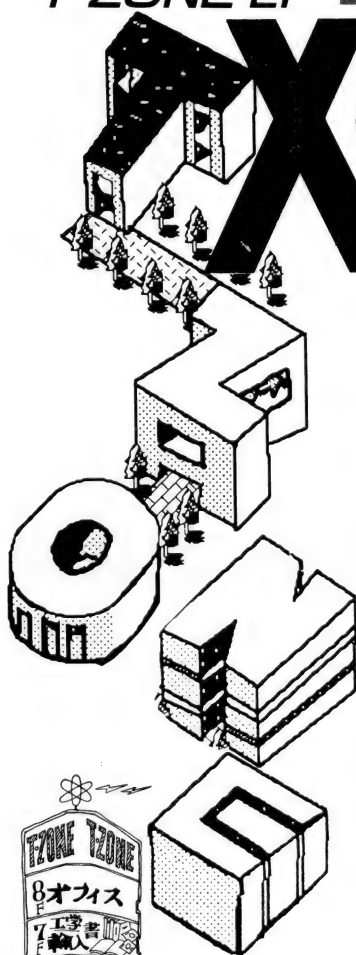
「オリент」「UC」「マスター」カードが1つになった。
「ボーナス一括払い」OK! 「通信販売」も
お手軽にご利用頂けます。そのほか、便利でお得な
特典がいっぱい! 今がチャンス!!
詳しくは、店頭にてどうぞ!!

68000をトータルサポート

T-ZONE 2F

SHARP Authorized.....

X68000
PRO SHOP



68000

新ラインナップ大好評発売中!!

EXPERT II・EXPERT II HD

PRO II・PRO II HD・SUPER HD

■新高速BIOS、SX-Window等、期待のX68000
新シリーズを全品超特価販売中です。
もちろんクレジットもOK!
X68000シリーズはT-ZONEにおまかせ下さい。

T-ZONE正社員・長期アルバイト募集中!

☆お問い合わせは総務課鈴木まで(TEL 03-257-2630)

T-ZONE

営業時間: AM10:30~PM7:00

下記T-ZONE各店でも扱っています。

宇都宮店: ☎0286(63)4949 川口店: ☎0482(68)7826 静岡店: ☎0542(83)1331 横浜店: ☎045(641)7741

大宮店: ☎048(652)1831 東ラジ店: ☎03(257)2694 パーツショップ: ☎03(257)2655

●マイコン通販利用の方へ: 現金留保で送金される際は、住所・氏名・TEL番号・希望商品名(詳しく)を明記して下さい。振込を御希望の方は下記銀行へお願いします。
(尚、いずれも予めTELにて、御予約・送料確認の上御送金下さい) (振込口座: 埼玉銀行 秋葉原支店 当座2705 埼玉電子工業)

☆この広告の提示価格には、消費税は含まれておりません。

OS-9 / 68000

☐OS9/68000 (SHARP) ¥29,800
☐C & PRO PACK (マイクロウェア) ¥58,000
☐MW-BASIC (マイクロウェア) ¥60,000
☐BTree09 (ARK) ¥36,000
MW-BASIC用のISAM用B-Treeパッケージです。応用例として住所録と販売管理プログラムが付属。全ソースコード付です。
(このソフトを動かすためにはMW-BASICが必要です。)

☐UD-CACHE (ARK) ¥16,000
すべてのRBFデバイスに対応するキャッシュです。

☐FBU (ARK) ¥38,000
ハード・ディスクバックアップユーティリティです。応用例として住所録と販売管理プログラムが付属。全ソースコード付です。
(このソフトを動かすためにはMW-BASICが必要です。)

☐VSED (FORKS) ¥28,000
OS9/68000で唯一オートバックアップをサポートしたスクリーンエディタです。

ニューリリースソフトウェア

CSG IMS V2.0
Clearbrook Software Group Information Management System

株星光電子
¥118,000

あのCSG-IMSがX68000にリリースされました。高度の処理に対応可能な言語型リレーショナルデータベースです。フォーマットを利用して簡単にシステム設計を行えます。OS9の特長を活かして、リアルタイム、マルチタスク、マルチユーザが必要なアプリケーションを構築することが可能です。もちろん、Cやアセンブラのモジュールを呼び出すことも可能、OAはもちろんFAにも対応できるバリエーションが豊富です。

OS9-SHL FORKS

Super Shell for OS9/68K ¥12,800

お待ちかね、OS9-SHLがX68000に対応しました。

■プロシージャファイルで引数、リダイレクト、パイプが使用可能。
■環境変数を数値または文字列として演算可、制御文で使用できます。
■豊富な制御文。While, wend, loopout, continue, if, else.....
■ヒストリやエイリアスをサポート、マクロコマンドもOK。
■カーソルキーに対応。

C SRC DBG マイクロウェアシステムズ

C SOURCE LEVEL DEBUGGER ¥39,800

OS9/X68000用Cソースレベルデバッガ発売開始!!
これでデバッグも楽々。

Oh! FMコーナー

FMシリーズ用OS9に新ソフト登場!

日本ソフトバンク

DB-09 (FM-7.77, AV.11) ¥18,252 /

OS9上で走るリレーショナルデータベースマネージャーです。問い合わせ形式で取扱い簡単。
なんとCによる全ソース付。

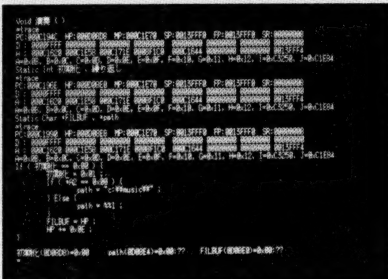
近日発売
10,000円

プログラム オペレーティング システム

バッチ処理の手軽さと、C言語ライクな
制御コマンドで、プログラムをチェー
ンする新しいタイプのインタプリタです。

デバッグ

デバッグの無い
プログラミング
なんてどうして
も考えられ無い
からデバッグの
環境も大切にし
ました。



あらたに、プログラム相互の通信機能が加わり、複数のプログラ
ムの間でメッセージをかわしながら、お互いを協調させることが
可能になりました。

これにより、今まで複雑であった条件による同一プログラムのと
りこみを避けることが、より簡単に定義できるようになりました。
また、このメッセージはチャイルドプロセスにも引き継がれるた
め、コンパイル時だけでなくプログラムの実行時にも生かされ、
さらに、プログラムの可能性が大きく広がりました。

No. 10



好評
発売
中

△68000専用
多機能デジタルサウンドツール

DiSS-P

ディスピー

Digital Sound System

豊富な機能をギッシリツメて、7,800円で登場!!

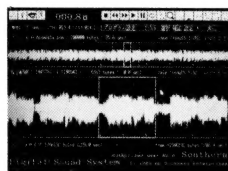
新時代の録音・編集・再生システム登場!

X68000専用に開発・設計しそのハイスペックを
継承し、持つ機能を最大限に活用した、新しい時
代の幕開けにふさわしいディスピーの誕生です。

特長

- すべてのサウンドをそっくりデジタル録音
ディスピー独自の長時間録音はナレーションからミュージッ
クにいたるまであらゆるニーズに対応
- 波形編集でプロフェッショナルなサウンドクリエイト
波形を確認しながら簡単なマウス操作でオリジナルサウンド
をワンタッチでアレンジ

- ワンタッチ再生やプログラム再生など多彩な再生機能
 - X68000が自在にしゃべる、スピーチ機能
 - 新時代のメール、ボイスメールシステム
 - データは自作プログラムにそのまま利用可能
 - ハイスピードなデータ処理とグラフ表示
 - 誰でも楽しめる豊富な音声データ付属
 - 買ったその日から使えるイージーオペレーション
 - X68000が再生できるすべてのデータの編集が可能
- ※この他機能満載、使い方もいろいろ、実用性を意識した仕
様です。お気軽にお問合せください。
※改良のため、内容の一部を予告なく変更することがあります。



(※写真は1M増設時です)

通信
販売

画面に皆様のお名前をお入れしてお届けします。住所・氏名
ふりがなを明記し7,800円を、現金書留・郵便振替・銀行振込
の何れか下記宛にお願いします。(税込み・送料サービス)
郵便振替 東京 8-404042 サザンエンタープライズ
銀行振込 三和銀行 往來支店 当座 308061

サザンエンタープライズ

〒142 東京都品川区戸越5-12-17 TEL・FAX 03-787-3932

《広告の半ページ》世界人類が断么でありますように

月刊 電脳倶楽部

90年7月号 (Vol.26)

6月18日発送

2HDディスクに入ったX68000のための雑誌だっ!

100の階乗も一発計算

多倍長整数 演算パッケージ

SRAMの留守番役

TALKON.X

ラベル作成支援ツール

MKNAME.X TXNAME.X HCPY1024.X

その他、便利なツール、ビーブ音、読み物などを満載!

(なお、内容は一部変更されることがあります。ご了承下さい)

編集長祝一平からの御挨拶「どーもすいません。どうしてもいっぺんやってみたかったネタなんです。ヘコヘコ」

満開製作所 電脳倶楽部 編集部

〒171 東京都豊島区要町1-19-3 いさみビル4F
TEL.(03)554-9282/FAX.(03)554-3856

販売方法は通信販売のみです。お申し込みの方法は左記の住所へ現金書留で
定期購読 6ヶ月分 6,000円 (消費税込・郵送料サービス)
●6月18日以降に受け付けた分は、原則としてVol.26から発送します。新たに購読を
希望される方は、「新規」と御明記下さい。
●郵便振替を御利用の場合は口座番号「東京5-36284 満開製作所」をお願いいたします。
製品の性格上、返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返します。
(ご注意: バックナンバーの受け付けは、定期購読の方に限らせていただきます)

エミュレータ

好評発売中

定価¥9,800



X1エミュレータはX68000上でX1シリーズのアプリケーションを実行するためのソフトエミュレータです。X1のアプリケーションを完全にソフトウェアのみでエミュレートしているため、X1上での実行速度と比較して、平均3~5倍程度おそくなりますが、X68000のマシン上に実現した仮想X1マシンを楽しめます。また、X1とX68000の相互間でファイルを転送するためのユーティリティと専用ケーブルが付属しますので、X1上で作り上げたソフトの資産をX68000上に移行することも簡単にできます。

エミュレータの機能

- X1エミュレータはX1に相当する機能をエミュレート。
この仮想コンピュータには最大4つのドライブが仮想的に接続。
- X1エミュレータからみたドライブはHuman68kのドライブ上にあるファイルで仮想的に実現。このファイルはX1用の5" 2Dディスクのイメージをファイル転送ユーティリティでまるごと転送したものです。
- X1エミュレータで仮想的に実現したX1は仮想ドライブから起動。
このため仮想ドライブ用ファイルには、X1を立ち上げるために必要なHuBASICやCP/Mなどのシステムプログラムが必要。
- X1エミュレータでは、X1の持つVRAMを含むメモリーイメージやZ80CPUを仮想的にソフトウェアで実現。

ファイル転送ユーティリティ

ディスク転送

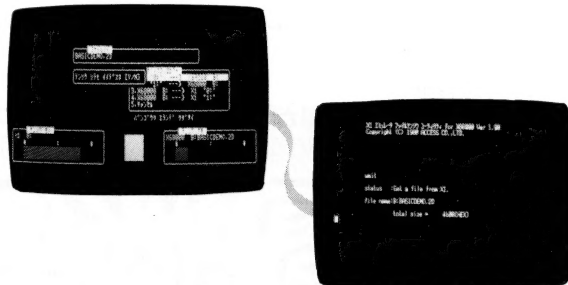
X1ディスク ↔ X68000 Human68k (5" 2Dディスクイメージファイル)

- X1エミュレータではHuman68k上のディスクイメージファイルを仮想ドライブとして使用。

ファイル転送

X1 BASIC: CP/M ↔ X68000 Human68k

- X1で作ったプログラム&データをX68000上で使用。
- ※ 付属の専用ケーブルをX1とX68000に接続してファイルを転送します。



エミュレータ Q&A

- Q. ファイル転送のために別途RS-232Cケーブルを買わないといけないのですか?
- A. 専用のケーブルが付属しますのでその必要はありません。
- Q. X1BASICのプログラムをX68000上のX-BASICで使えますか?
- A. 通常のセーブではコードが違うので使用できませんが、アスキーセーブしたファイルであればX-BASIC上でそのままロード可能です。
- Q. TurboBASICで作成した住所録などの漢字を含んだデータがあるのですがX68000上にファイル転送できますか?
- A. X1TurboもX68000も漢字はシフトJISコードなのでファイルの転送は可能です。ただし、漢字ROMを必要とするものはサポートしていません。

- Q. Turbo用のソフトは動きますか?

A. X1用のみでTurbo専用のソフトは動きません。

- Q. ゲームは動きますか?

A. 純粋にBASICでかかれたものは動きますが、プロテクトがかかったものや直接ハードをアクセスするような市販のゲームは動きません。

* タイミング等ハードウェアに依存するようなソフトは、原理上実行できない、もしくは正常に動作しない場合がありますのでご注意ください。
* 一部サポートしていない機能があります。

X1エミュレータ通信販売 購入希望として住所、氏名、電話番号をお知らせください。注文書をお送り致します。

発売中

X68000用

CONCERTO-X68K

MS-DOSエミュレータ

定価¥99,800

代理店募集

アクセスではこれらの製品の発売にあたり代理店を募集しております。詳しくはお問い合わせください。

* この商品価格には消費税は含まれておりません。

* MS-DOSはマイクロソフト社、CP/Mはデジタルリサーチ社の商標です。

* 文中のソフトウェアは各社の商標です。

* 製品の仕様、名称は予告なく変更する場合もございますのであらかじめご了承ください。

有限会社 アクセス 〒101 東京都千代田区神田神保町1-64
神保町協和ビル7F
TEL 03 (233) 0200(代) FAX 03 (291) 7019

おさがせ村サリー (ジャンプコード SAWAGASE)



「何でもまかせろっ!! みんなが集う、自由の広場」

「ミートでっせ!」の呼びかけに即座「参加っ!」と嬉しい返事。サリーのオフミはいつも大入り満員。初対面さんとも挨拶済めば、すぐにわきあいあい。まるで旧知の間柄。小さなミートも含めると毎週やっているかな? ワハハ、ヤリスギッテ? でも「バツ通は職種の壁を越えるっ!」と、オフミを重ね、メンバー間の親近度も抜群だから、MSG(メッセージ)の量はダントツ。オチャラケから専門的なことまでMSGの山、山、山。なんせ、パワーライター養成所もやってるくらいですわ。ハイ。チャレンジ精神も旺盛。現状の「テキストでしか意志を表現できない状態」の打破に挑戦して、新しい通信の世界へ…… というウケで4月より画像通信専用ボード「サリーギャラリー」をオープン。ここでは画像通信を勉強しながら、メンバーの描いたイラストやMEETの写真をアップして、メンバー各位のコミュニケーションを図っています。

その他 楽しいメニューがまだまだいっぱい!

- ★J&Pならではのパソコン・家電製品
の会員割引もあるONLINE SHOPPING。
- ★J&Pだから強い!! パソコン情報をはじめとする
役に立つDATA BASE。
- ★みんなでおしゃべりオンライントーク(CHAT機能)。
- ★地域別・テーマ別ボードで充実のBBS(電子掲示板)。
- ★ビジュアルデータもばっちり送受信できるX-MODEM。

J&P HOT LINEへのご入会はスタータキットで。

買ったその日から
2週間無料で
アクセスできます。

お求めは、下記のお店へ。又は現金書留にて、¥3,000+¥90(消費税3%)=¥3,090を事務局までお送り下さい。
すぐにスタータキットをお送りします。

お問い合わせは 〒556 大阪市浪速区日本橋西1-6-5 上新電機株式会社
J&P HOT LINE事務局宛 TEL.(06)632-2521

スタータキットのお求めはJ&P各店でどうぞ。

渋谷店 東京都渋谷区道玄坂2丁目28番4号 ☎(03) 496-4141
町田店 東京都町田市森野1丁目39番16号 ☎(0427) 23-1313
八王子店 東京都八王子市旭町1番1号八王子そうじ方 ☎(0426) 26-4141
立川店 東京都立川市幸町4-39-1 ☎(0425) 36-4141
厚木店 厚木市中町3-4-3 ☎(0462) 25-1548
富士店 富士市桜町2-1-10 ☎(0764) 32-3133
金沢店 金沢市入江2-63 ☎(0762) 91-1130
寺地店 金沢市寺地2-3 ☎(0762) 47-2524
大須店 名古屋市中区大須4丁目2-48 ☎(052) 262-1141

テクノランド 大阪市浪速区日本橋5丁目6番7号 ☎(06) 634-1211
メディアランド 大阪市浪速区日本橋5丁目8番26号 ☎(06) 634-1511
コスモランド 大阪市浪速区難波中2丁目1番17号 ☎(06) 634-3111
U.S.LAND 大阪市浪速区日本橋4丁目9番15号 ☎(06) 634-1411
ビジネスランド 大阪市北区梅田1-1-3大阪駅前第3ビルB2 ☎(06) 348-1881
梅田店 大阪市北区小松原町1-10 ☎(06) 362-1141
高槻店 高槻市高槻町11番16号 ☎(0726) 85-1212
くすは店 枚方市楠葉花園町15番2号 ☎(0720) 56-8181
千里中央店 豊中市千里東町1-3SENCHU PAL2番街4F ☎(06) 834-4141
摂津富田店 高槻市大畑町24-10 ☎(0726) 93-7521
寝屋川店 寝屋川市緑町4-20 ☎(0720) 34-1166

藤井寺店 藤井寺市岡2丁目1番33号 ☎(0729) 38-2111
岸和田店 岸和田市土生町2451-3 ☎(0724) 37-1021
さんのみやばん 神戸市中央区八幡通3-2-16 ☎(078) 231-2111
西宮店 兵庫県西宮市河原町5-11 ☎(0798) 71-1171
姫路店 姫路市東延1丁目1番住友生命姫路南ビル1F ☎(0792) 22-1221
京都寺町店 京都市下京区寺町通仏光寺下町恵美須町54 ☎(075) 341-3571
京都近鉄店 京都市下京区烏丸通七条下町東塩小路702 ☎(075) 341-5769
和歌山店 和歌山市元寺町4丁目4番地 ☎(0734) 28-1441
奈良1ばん館 奈良市三条町478-1 ☎(0742) 27-1111
郡山インター店 大和郡山市横田693-1 ☎(07435) 9-2221
熊本店 熊本市手取本町4-12 ☎(096) 359-7800

ADVANCED TURBO

先駆の“Z”アビリティがパソコンクリエイターを魅了する。



AV1 パソコンテレビ **turbo Z III**

パーソナルコンピュータ+キーボード+マウス	CZ-888C-BK 標準価格 169,800円(税別)
14型カラーディスプレイテレビ	CZ-860D-BK 標準価格 92,200円(税別)
チルトスタンド	CZ-6ST1-B 標準価格 5,800円(税別)

クリエイティブマインドを刺激するAV機能 テレビ、ビデオ、ビデオディスクなどの映像を最大4,096色のリアルな画像で瞬時にグラフィック画面に取り込めるカラー画像デジタイズ機能を標準装備。4段階の量子化取り込み、42通りのモザイク取り込みなど多彩なトリック取り込み処理もサポート。さらにクロマキー合成、インターレーススーパーインポーズ、4,096色対応デジタルテロップ機能、ステレオFM音源…先駆のAV機能がアートワークの領域をさらに広がります。

AV指向の高水準ベーシックZ-BASIC搭載 多色グラフィック、カラー画像処理、ステレオFM音源、バンクメモリ対応など、ターボZシリーズが本来もつクリエイティブな機能をフルサポート。また豊富な画面モードで多色を駆使するときに便利なグラフィック用関数(HSV、RGB、HALF、CDOWN、CUP)も装備。さらにFM音源制御用ステートメントとしてX68000と命令コンパチの拡張MMLの採用によりスムーズな8音同時演奏を実現しています。

●メインメモリ128Kバイト標準装備、Z-BASICで最大576Kバイトまでサポート●1Mバイトの5インチフロッピーディスクドライブ2基搭載●JIS第1/第2水準漢字、「システム・ユーザー辞書」を標準装備した高度な日本語処理機能●ニューデザインのマウス標準装備●X1ターボシリーズの豊富なソフト資産が活用できるコンパチブル設計●プリンタ、RS-232Cなど豊富なインターフェイスを装備●ドットピッチ0.39mmのハイコントラストブラウン管、15kHz/24kHzのデュアルスキャン方式採用14型カラーディスプレイテレビ(別売)。